R Package, Multivariate p-Value Estimation: MANOVA,Block Independence & Bartlett's M ECM Test,Plus 1-Way MANOVA Study Design

James V. Chavern

2018-01-21

Introduction:

Name

The name of this R package is wmpvaer. This stands for "Wilks MANOVA p-Value Alternative Estimation". The first part of this package contains R programs implementing an alternative to the approximate F Statistic p calculation (used in the R MANOVA package) for the Wilk's statistic used in MANOVA with a calculation developed in 2005 by Butler and Wood. The package also supplies an S4 object so that the same statistics can be computed with other R programs other than R MANOVA.

The second part of this package computes p-values for a likelihood ratio statistic of block independence, implementing another method from Butler and Wood.

The third part of this package computes p-values for Bartlett's M test for Equality of Covariance matrices (ECM), again from Butler and Wood's 2005 paper.

The fourth part of this package computes sample size estimates for 1-Way MANOVA analysis using Wilks Lambda and saddlepoint approximation.

PART 1 : MANOVA,Wilks Lambda and the Wishart Distribution

The distribution of the Wilks Lambda is the ratio of two independently distributed Wishart random variables. It is so complicated (intractable) that its expression in real numbers is almost never used. Currently approximate F tests are the only widely distributed tests for estimating p values for an observed Wilks Lambda statistic.

There is another approach to approximating the Wilks Lambda distribution and p values associated with it. The moment (and cumulant) generating functions of Wilks Lambda distribution have been known since 1963. These express the Wilks Lambda distribution (on a one to one basis) in terms of complex numbers instead of real numbers. Once either the moment or cumulant generating function of a probability distribution are known saddlepoint approximation becomes feasible.

The idea of saddlepoint approximation originated with Henry Daniels in 1954 ("Saddlepoint approximations in statistics." Annals of Mathematical Statistics, v. 25 ,pp. 631-650,1954). In 1980 Lugannani and Rice ("Saddlepoint Approximation for the Distribution of the Sum of Independent Random Variables" in Advances in Probability, Vol. 12, pp. 475-490 ,1980) extended Daniels method to the estimation of cumulative distribution functions.

In turn in 2005, in a paper written by Butler, R.W. and Wood, A.T.A. ("Approximation of Power in Multivariate Analysis" in Statistics and Computing Vol. 15, pp. 281-287) Butler and Wood explained how the methods developed could be applied to the otherwise intractable Wilks Lambda distribution. The system

of equations, which translates the complex number representation of the Wilks Lambda distribution into a real number approximation, are lengthy and involved. However, as this package wmpvaer demonstrates, they can be computed. And, with the use of the R high precision computation package Rmpfr, result in an accurate representation of the Wilks Lambda Distribution, which is in turn used to compute accurate estimates of the power of a Wilks Lambda statistic. When used retrospectively (after data is collected), "power" has a one to one relationship to p-values.

wmpvaer p-values created differ substantially from the R MANOVA approximations in the datasets examined

One of the statistician W. Edwards Deming's aphorisms was that "There is no true value of anything". All that exists, Deming would say, are methods for deriving answers and the answers themselves. When the methods used differ so do the answers arrived at, in general.

Butler and Wood's p approximations are derived, mathematically, directly from the Wilks distribution itself. There is no intermediate approximation requiring the F distribution. Further, there are no justifications or arguments required as to whether the F distribution matches the distribution of the ratio of two independently distributed Wishart random variables.

This vignette does not discuss Butler/Wood in detail

If you want to understand the methods underlying the Butler/Wood equations, it is best read Butler and Wood's paper. This vignette primarily concerns samples of data and R coding. The part of the Butler/Wood equations I code with this package addresses only a small portion of the applications possible (p-values and 1-way MANOVA design/sample size estimation).

Samples of Data and R Coding : `wmpvaerf`

With this package, I include three datasets named respectively "plastic", "soils", and "wolf". Their origins are documented in this package but all three are standard datasets used to demonstrate MANOVA analysis.

The function used to derive the Wilks power estimates and p values, using output from the R MANOVA routine, is called `wmpvaerf`. The first matrix output is a matrix composed of the Wilks values and degrees of freedom generated by R's basic MANOVA program using the `summary(fit, test="Wilks")$stats` option and the log of this value. `wmpvaerf` also uses the R MANOVA matrix created using `summary(fit)$Eigenvalues`, but this matrix is not included in the wmpvaerf output.

The 2nd matrix of `wmpvaerf` output are the values of power and the p-values computed with 120 bit precision ("quadruple precision").

The 3rd matrix of `wmpvaerf` output reformates the second for convenience and adds the values of "r-manova˜p" derived from the R MANOVA program.

Example 1: Plastic Dataset

Using the "plastic" dataset (also used as an example in R's MANOVA function documentation), the output of `wmpvaerf` is:

```
suppressPackageStartupMessages(library(Rmpfr));
suppressPackageStartupMessages(library(doParallel));
library(wmpvaer);
# prepare data
```

```
     data(plastic);
     Y<-as.matrix(plastic[,1:3]);
     rate<-matrix(data=(plastic$rate),nrow(plastic),1);
     additive<-matrix(data=(plastic$additive),nrow(plastic),1);
     # run wmpvaer function
     wmpvaerf(Y~ (rate*additive));
[[1]]
              Df  Wilks              ln(Wilks)
rate          1  0.381858384661142 -0.962705459891849
additive      1  0.523034895418919 -0.648107095500711
rate:additive 1  0.777105757873425 -0.252178827357477
Residuals     16

[[2]]
              Pr(x>ln(Wilks)),Power,precBits=120
rate             0.99999999999999416012239272263966343901
additive         0.99999999999999999976662838952759483172
rate:additive 4.0327993146447476244262410758493742103e-28
              Pr(x<=ln(Wilks)),p Value,precBits=120
rate          5.8398776072773603365609934757202060454e-15
additive      2.3337161047240516827595468766170403915e-19
rate:additive    0.99999999999999999999999999959672006857

[[3]]
              Power,formatted  p,formatted   r-manova~p
rate              1.000000    5.839878e-15  0.003034045
additive          1.000000    2.333716e-19   0.02474528
rate:additive 4.032799e-28        1.000000    0.3017816
```

Example 2: Wolf Dataset

This a dataset of 9 measurements of 25 wolf skulls from different locations in North America and of different sexes. This data was gathered in the 1950s through the 1970s.

Using all 9 measurements, the following R code results in the following output:

```
     suppressPackageStartupMessages(library(Rmpfr));
     suppressPackageStartupMessages(library(doParallel));
     library(wmpvaer);
     # prepare data
     data(wolf) ;
     location<-matrix(data=(wolf$location),nrow(wolf),1);
     sex<-matrix(data=(wolf$sex),nrow(wolf),1);
     wolf$sex<-NULL;
     wolf$location<-NULL;
     wolfm1<-as.matrix(wolf[,1:9]);
     # run wmpvaer function
     wmpvaerf(wolfm1 ~  (sex * location));
[[1]]
             Df  Wilks              ln(Wilks)
sex          1  0.149157496173264  -1.90275250999326
location     1  0.0475427371985173 -3.0461262419613
sex:location 1  0.350558699882641  -1.04822711182898
```

```
Residuals    21

[[2]]
             Pr(x>ln(Wilks)),Power,precBits=120
sex          1.485732459078273210465826083540365304 1e-141
location     5.017730723008527095268077224670114610 6e-111
sex:location 8.531407145269148711120757427921720822 4e-182
             Pr(x<=ln(Wilks)),p Value,precBits=120
sex                                                  1
location                                             1
sex:location                                         1

[[3]]
             Power,formatted  p,formatted    r-manova~p
sex          1.485732e-141        1.000000  0.0004375614
location     5.017731e-111        1.000000   3.623591e-7
sex:location 8.531407e-182        1.000000    0.05238652
```

Example 3: Wolf Dataset

Analysis with only the first three dependent variables in the wolf skull measurement dataset ("palatal (roof of the mouth) length", "post palatal length" and "zygomatic (cheek bone) width") gives the following results:

```
     suppressPackageStartupMessages(library(Rmpfr));
     suppressPackageStartupMessages(library(doParallel));
     library(wmpvaer);
     # prepare data
     data(wolf) ;
     location<-matrix(data=(wolf$location),nrow(wolf),1);
     sex<-matrix(data=(wolf$sex),nrow(wolf),1);
     w<-c(wolf$palatal_L, wolf$post_p_L,wolf$zygomatic_W);
     wolfm1<-matrix(data=w,nrow(wolf),3);
     # run wmpvaer function
     wmpvaerf(wolfm1~sex*location);
[[1]]
             Df  Wilks             ln(Wilks)
sex          1  0.266386005753227 -1.32280887265206
location     1  0.13287464892308  -2.01834908448284
sex:location 1  0.669209994037468 -0.401657375675144
Residuals    21

[[2]]
             Pr(x>ln(Wilks)),Power,precBits=120
sex                0.99999999999999999999994543587757341
location        0.99999999999999999999999999486772659496
sex:location 2.898844525014046005221240713822206209 6e-32
             Pr(x<=ln(Wilks)),p Value,precBits=120
sex          5.456412242658999535764586324426933250 7e-23
location     5.132273405038837705417832829967094182 3e-27
sex:location    0.99999999999999999999999999999997101175

[[3]]
             Power,formatted  p,formatted   r-manova~p
```

```
sex              1.000000    5.456412e-23  1.097254e-5
location         1.000000    5.132273e-27  1.594996e-8
sex:location 2.898845e-32        1.000000   0.04985222
```

Example 4: Soils Dataset

Using the soils dataset and the following code:

```
     suppressPackageStartupMessages(library(Rmpfr));
     suppressPackageStartupMessages(library(doParallel));
     library(wmpvaer);
     # prepare data
     data(soils) ;
     soilsfactors<-soils[,1:3];
     block<-as.factor(soilsfactors[,1]);
     contour<-soilsfactors[,2];
     depth<-soilsfactors[,3];
     soilsm1<-as.matrix(soils[,4:12]);
     # run wmpvaer function
     wmpvaerf(soilsm1~  (block+contour*depth));
[[1]]
              Df  Wilks                ln(Wilks)
block         3  0.0794302001420071  -2.53287662861212
contour       2  0.0658176778549351  -2.72086681632426
depth         3  0.00496311671124768 -5.3057213663834
contour:depth 6  0.216610557064923   -1.52965420569107
Residuals     33

[[2]]
              Pr(x>ln(Wilks)),Power,precBits=120
block         4.8389446118168200565958000805133477743e-233
contour       2.7596250592977486862071520800115839129e-208
depth                                                     1
contour:depth 2.6819978668486090092069178655859215868e-327
              Pr(x<=ln(Wilks)),p Value,precBits=120
block                                                     1
contour                                                   1
depth                                                     0
contour:depth                                             1

[[3]]
              Power,formatted  p,formatted    r-manova~p
block         4.838945e-233        1.000000   3.346773e-6
contour       2.759625e-208        1.000000   2.452713e-9
depth             1.000000         0.000000  1.770503e-19
contour:depth 2.681998e-327        1.000000     0.7392245
```

If you use all 48 observations and all 9 dependent variables, the power analysis indicates that depth distinguishes the dependent variables. This less than surprising result is that, based on this dataset, depth separates the soil types, alone. However, if you had only R MANOVA approximate p-values to consider you would come to a different conclusion.

Example 5: Soils Dataset, Using Only The First 7 Dependent Variables

The soils dataset but use only the first 7 dependent variables and the following code:

```
     suppressPackageStartupMessages(library(Rmpfr));
     suppressPackageStartupMessages(library(doParallel));
     library(wmpvaer);
     # prepare data
     data(soils) ;
     soilsfactors<-soils[,1:3];
     block<-as.factor(soilsfactors[,1]);
     contour<-soilsfactors[,2];
     depth<-soilsfactors[,3];
     soilsm1<-as.matrix(soils[,4:10]);
     # run wmpvaer function
     wmpvaerf(soilsm1~  (block+contour*depth));
[[1]]
              Df  Wilks              ln(Wilks)
block         3  0.137862375511398 -1.98149937034131
contour       2  0.09888854973972  -2.31376182319639
depth         3  0.029714683116699 -3.51611397430589
contour:depth 6  0.27522379287536  -1.29017071999092
Residuals     33

[[2]]
              Pr(x>ln(Wilks)),Power,precBits=120
block         5.079372784681287870427877708809225841 8e-162
contour                                                    1
depth                                                      1
contour:depth 9.275660907836654390794413153101705509 2e-224
              Pr(x<=ln(Wilks)),p Value,precBits=120
block                                                      1
contour                                                    0
depth                                                      0
contour:depth                                              1

[[3]]
              Power,formatted  p,formatted    r-manova~p
block         5.079373e-162        1.000000   1.314759e-5
contour           1.000000         0.000000   3.459357e-9
depth             1.000000         0.000000  2.662329e-13
contour:depth 9.275661e-224        1.000000     0.5136871
```

If you use all 48 observations and and only the first 7 dependent variables,contour appears as significant in contrast to example 4. The two variables dropped were "Exchangeable+soluble Na" (Na is Sodium, Exchangeability or CEC is a measure of soil fertility) and "Conductivity" (waters ability to conduct electricity-pure distilled water is a bad conductor).

Samples of Data and R Coding : `mlandrf`

To use the estimation routines with programs other than R MANOVA, an S4 object `mlandr` was written. This object can be used with any routine that analyzes models with multiple dependent variables and that can produce E (an error sum of squares matrix) and H (a sum of squares matrix due to an hypothesis). If

E and H exist, the matrix E-inverse*H can then be created and the eigenvalues of this matrix found.

Example 1 below uses E-inverse*H eigenvalues from the plastic dataset (these are stored in workspace eigs_plastic with eigenvalues generated by R MANOVA). Example 2 creates the MANOVA analysis using the R function lm( ) instead. The `mlandr` S4 object is executed using the R function `mlandrf`.

Example 1: Eigs_plastic Dataset

Using "eigs_plastic" dataset (eigenvalues derived from E-Inv*H),output of mlandrf is:

```
     suppressPackageStartupMessages(library(Rmpfr));
     suppressPackageStartupMessages(library(doParallel));
     library(methods);
     library(wmpvaer);
     # prepare data
     data(eigs_plastic);
     eigsm<-as.matrix(eigs_plastic)
     colnames(eigsm)<-c(" "," "," ");
     rownames(eigsm)<-c("rate","additive","rate:additive");
     e_dfm<-16;
     m_dfm<-c(1,1,1);
     # run S4 object "mlandr" wrapper function
     mlandrf(e_dfm,m_dfm,eigsm);
[[1]]
              Df    Wilks       ln(Wilks)  p,formatted
rate          1  0.381858382 -0.962705467 5.840229e-15
additive      1  0.523034902 -0.648107084 2.333694e-19
rate:additive 1  0.777105780 -0.252178799     1.000000
Residuals     16

[[2]]
                   Pr(x>ln(Wilks)),Power,precBits=120
rate              0.99999999999999415977111235584492266749
additive           0.99999999999999999976663064844033281 31
rate:additive 4.033219288386158355501604210650855307 4e-28
                   Pr(x<=ln(Wilks)),p Value,precBits=120
rate          5.840228887644155077332506745365619098 4e-15
additive      2.333693515596671869008810654572731251 1e-19
rate:additive    0.99999999999999999999999999959667807126
```

The computed Wilks statistics are slightly different from the values produced by R MANOVA. They are computed with 120 bit precision from the derived eigenvalues.

Example 2: Plastic Dataset, MANOVA analysis using `lm( )` function and `mlandrf`

Use lm( ) function, create MANOVA w "plastic" dataset & derive eigenvalues from "E-Inv*H"s :

```
     suppressPackageStartupMessages(library(Rmpfr));
     suppressPackageStartupMessages(library(doParallel));
     library(wmpvaer);
     library(methods);
     # prepare data
     data(plastic);
     Y<-as.matrix(plastic[,1:3]);
```

```
     rate<-matrix(data=(plastic$rate),nrow(plastic),1);
     additive<-matrix(data=(plastic$additive),nrow(plastic),1);

     # run lm( )
     fit <-lm( Y ~  (rate*additive));

     #compute m_df & e_df
     mstats<-summary(manova(fit), test="Wilks")$stats;
     fm<-as.matrix(mstats[,1]);
     nrms<-nrow(fm);
     m_df<-fm[1:(nrms-1),];
     m_dfm<-as.matrix(fm[1:(nrms-1),]);
     e_df<-fm[nrms,];

     # get list of Sum of Squares matrices
     ssv<-summary(manova(fit))$SS
     m1<-length(ssv);

     # first,derive E-Inv
     em1<-ssv[[m1]]; #the last SS matrix in the ssv list is the residual matrix
     e_inv<-solve(em1); # inverts the em1 matrix

     # second, compute eigenvalues for an eigenvalue matrix
     eigsm<-t(as.matrix(eigen((e_inv %*% ssv[[1]] ))$values))
     if(m1>2){
     for (i in (2:(m1-1))) {
     eigenvs<-t(as.matrix(eigen((e_inv %*% ssv[[i]] ))$values))
     eigsm<-rbind(eigsm,eigenvs);
     }
     }
     colnames(eigsm)<-c(" "," "," ");
     rownames(eigsm)<-rownames(m_dfm);

     # run S4 object "mlandr" wrapper function
     mlandrf(e_df,m_df,eigsm);
[[1]]
              Df    Wilks       ln(Wilks)  p,formatted
rate          1  0.381858385 -0.962705460 5.839878e-15
additive      1  0.523034895 -0.648107096 2.333716e-19
rate:additive 1  0.777105758 -0.252178827     1.000000
Residuals     16

[[2]]
                   Pr(x>ln(Wilks)),Power,precBits=120
rate              0.99999999999999416012239272262867868165
additive          0.99999999999999999976662838952759471662
rate:additive 4.032799314644743221805800364275776083 5e-28
                   Pr(x<=ln(Wilks)),p Value,precBits=120
rate          5.83987760727737132131834609067893171 97e-15
additive      2.33371610472405283380361520180096819 45e-19
rate:additive    0.99999999999999999999999999959672006857
```

The computed Wilks statistics are slightly different from the values produced by R MANOVA. They are computed with 120 bit precision from the derived eigenvalues.

PART 2 : p-Values for LR Test of Block Independence

What this problem is.

Let x_i be a numerical vector with values assumed to be normally distributed random variables. Further, assume there are N of these vectors, that is, $x_1, \cdots, x_N$. These could easily be placed in a matrix with N columns.

Suppose you divide the matrix into two blocks,with the number of columns in the first block=p_1 and the number of columns in the second=p_2. p_1 and p_2 are restricted as follows: $p_1 + p_2 = N$ and $p_1 <= p_2$.

From this information compute a sample variance-covariance matrix for all the data $A = \widehat{\Sigma}$ and a sample variance-covariance matrix $A_{1,1} = \widehat{\Sigma_{1,1}}$ for block 1 and $A_{2,2} = \widehat{\Sigma_{2,2}}$ for block 2. Further, compute a sample cross covariance matrix $\widehat{\Sigma_{1,2}} = \widehat{\Sigma_{2,1}{}^T}$.

A test of whether block 1 was statistically independent from block 2 (that is, $\Sigma_{1,2} = (\Sigma_{2,1})^T = 0_{p1,p2}$) could be (Muirhead,R.J,1982 Aspects of Multivariate Statistical Theory, John Wiley,New York,Section 11.2), a likelihood ratio test, specifically:

$$\Lambda_{BI} = \frac{|A|}{(|A_{1,1}| * |A_{2,2}|)}$$

where $|\bullet|$ is the determinant of a matrix. The distribution of this statistic is a ratio of two independently distributed Wishart random variables, the noncentral version of which is determined by $n = N - 1$,p_1,p_2, and the estimated eigenvalues of the matrix $(\Sigma_{1,1})^{-1} * \Sigma_{1,2} * (\Sigma_{2,2})^{-1} * \Sigma_{2,1}$.

But even though the real number form of this distribution is known, it is intractable and not of much use. Butler and Wood showed how saddlepoint methods applied to the known moment generating function of the distribution of Λ_{BI} (along with the Lugannani and Rice approximation) converts the Λ_{BI} distribution into a usable real numbers form. This R package computes these methods with the functions `statsBIf` and `statsBIncatf`.

Butler and Wood did not provide some essential information for this computation in their paper. Specifically they did not provide the formula for the analytical derivative $\widehat{K\prime(s)}$ (as they did for the Wilks distribution). But this derivative and the derivative for statsECMf (below) were found and implemented with these programs.

What the computed distribution provides.

In the absence of a usable form of the Λ_{BI} probability distribution what this statistic Λ_{BI} has been used for is to, roughly, reject the null hypothesis $(\Sigma_{1,2} = (\Sigma_{2,1})^T = 0_{p1,p2})$. If Λ_{BI} is "small" then reject the null hypothesis. What is small or large is not defined (other than Λ_{BI} is between 0 and 1).

Once you can approximate the Λ_{BI} probability distribution, then you can derive a p-value. A small p-value estimate means that there is a small probability of rejecting the null hypothesis in error. A well defined reason to reject a null hypothesis is found.

It turns out that, in the datasets examined, the p-value often doesn't correspond to the size of the Λ_{BI} value. Small values of Λ_{BI} can have large p-values and large values of Λ_{BI} can have small p-values. Conclusions based on the raw value of Λ_{BI} alone don't have much of a rational basis.

Samples of Data and R Coding : `statsBIf` and `statsBIncatf`

Example 1: Plastic Dataset, statsBIf

This is a simple example of `statsBIf` using the plastic dataset. All that is needed for `statsBIf` to run is a matrix composed of numeric values and the definition of a variable identifying what columns are in the first block.

```
    suppressPackageStartupMessages(library(Rmpfr));
    library(wmpvaer);
    # prepare data
    data(plastic);
    plasticm<-as.matrix(plastic[,1:3]);
    block_v<-c(2)  # identify columns in block 1
    statsBIf(plasticm,block_v);
  c# name_blk1 c# name_blk2 lambdaBI   p value
  2  gloss     1  tear      0.96233358 0.9196664
               3  opacity
```

Example 2: Wolf Dataset, statsBIf

This is a more involved example using the wolf dataset. Again,all that is needed for `statsBIf` to run is a matrix composed of numeric values and the definition of a variable identifying what columns are in the first block. Since there are only 9 columns and p1 must be<=p2, 4 columns is the maximum size for block 1.

```
    suppressPackageStartupMessages(library(Rmpfr));
    library(wmpvaer);
    # prepare data
    data(wolf) ;
    wolf$sex<-NULL;
    wolf$location<-NULL;
    wolfm1<-as.matrix(wolf[,1:9]);
    block_v<-c(1,3,5,7)  # identify columns in block 1
    statsBIf(wolfm1,block_v);
  c# name_blk1      c# name_blk2        lambdaBI    p value
  1  palatal_L      2  post_p_L         0.023164421 4.5817539e-05
  3  zygomatic_W    4  palatal_W1
  5  palatal_W2     6  postg_foramina_W
  7  interorbital_W 8  braincase_W
                    9  crown_L
```

Example 3: Plastic Dataset, statsBIncatf

`statsBIncatf` uses the methods in `statsBIf` to produce analysis of all columns taken n at a time. Possible values of n are from 1 to floor(N/2), where N is the total number of columns in the data matrix. For the plastic dataset, with 3 columns, the only possible value of n is 1.

```
    suppressPackageStartupMessages(library(Rmpfr));
    suppressPackageStartupMessages(library(doParallel));
    library(wmpvaer);
    # prepare data
    data(plastic);
    plasticm<-as.matrix(plastic[,1:3]);
```

```
    nc_at_time<-1;
    statsBIncatf(plasticm,nc_at_time)
[[1]]
  c# name_blk1 c# name_blk2 lambdaBI   p value
  1  tear      2  gloss     0.97154632 0.92672565
               3  opacity

[[2]]
  c# name_blk1 c# name_blk2 lambdaBI   p value
  2  gloss     1  tear      0.96233358 0.9196664
               3  opacity

[[3]]
  c# name_blk1 c# name_blk2 lambdaBI   p value
  3  opacity   1  tear      0.99032795 0.94404996
               2  gloss
```

This analysis is interesting,though. It presents evidence that the three dependent variables in the data, "tear","gloss" and "opacity" all display independence from one another.

Example 4: Wolf Dataset, statsBIncatf

`statsBIncatf` uses the methods in `statsBIf` to produce analysis of all columns taken n at a time. Possible values of n are from 1 to floor(N/2), where N is the total number of columns in the data matrix. For the wolf dataset, with 9 columns, n could be from 1 to 4. However, 9 taken 4 would results in output with 126 elements, 9 taken 3, 84 elements, 9 taken 2, 36 elements. So in the interest of vignette conservation, this example shows only 9 columns taken 1 at a time in comparison with all other columns.

```
    suppressPackageStartupMessages(library(Rmpfr));
    suppressPackageStartupMessages(library(doParallel));
    library(wmpvaer);
    # prepare data
    data(wolf) ;
    wolf$sex<-NULL;
    wolf$location<-NULL;
    wolfm1<-as.matrix(wolf[,1:9]);
    nc_at_time<-1;
    statsBIncatf(wolfm1,nc_at_time)
[[1]]
  c# name_blk1 c# name_blk2        lambdaBI   p value
  1  palatal_L 2  post_p_L         0.13862872 0.053084376
               3  zygomatic_W
               4  palatal_W1
               5  palatal_W2
               6  postg_foramina_W
               7  interorbital_W
               8  braincase_W
               9  crown_L

[[2]]
  c# name_blk1 c# name_blk2        lambdaBI   p value
  2  post_p_L  1  palatal_L        0.15991961 1
               3  zygomatic_W
```

```
              4  palatal_W1
              5  palatal_W2
              6  postg_foramina_W
              7  interorbital_W
              8  braincase_W
              9  crown_L

[[3]]
  c# name_blk1   c# name_blk2        lambdaBI   p value
  3  zygomatic_W 1  palatal_L        0.17606424 0
                 2  post_p_L
                 4  palatal_W1
                 5  palatal_W2
                 6  postg_foramina_W
                 7  interorbital_W
                 8  braincase_W
                 9  crown_L

[[4]]
  c# name_blk1  c# name_blk2        lambdaBI  p value
  4  palatal_W1 1  palatal_L        0.3088967 0.77180576
                2  post_p_L
                3  zygomatic_W
                5  palatal_W2
                6  postg_foramina_W
                7  interorbital_W
                8  braincase_W
                9  crown_L

[[5]]
  c# name_blk1  c# name_blk2        lambdaBI   p value
  5  palatal_W2 1  palatal_L        0.23768878 0.99253059
                2  post_p_L
                3  zygomatic_W
                4  palatal_W1
                6  postg_foramina_W
                7  interorbital_W
                8  braincase_W
                9  crown_L

[[6]]
  c# name_blk1        c# name_blk2      lambdaBI   p value
  6  postg_foramina_W 1  palatal_L      0.19112488 0.0040741843
                      2  post_p_L
                      3  zygomatic_W
                      4  palatal_W1
                      5  palatal_W2
                      7  interorbital_W
                      8  braincase_W
                      9  crown_L

[[7]]
  c# name_blk1      c# name_blk2        lambdaBI   p value
```

```
  7  interorbital_W 1  palatal_L        0.25668665 0.97229474
                    2  post_p_L
                    3  zygomatic_W
                    4  palatal_W1
                    5  palatal_W2
                    6  postg_foramina_W
                    8  braincase_W
                    9  crown_L

[[8]]
  c# name_blk1   c# name_blk2        lambdaBI   p value
  8  braincase_W 1  palatal_L        0.59218481 0.85274185
                 2  post_p_L
                 3  zygomatic_W
                 4  palatal_W1
                 5  palatal_W2
                 6  postg_foramina_W
                 7  interorbital_W
                 9  crown_L

[[9]]
  c# name_blk1 c# name_blk2        lambdaBI   p value
  9  crown_L   1  palatal_L        0.55689771 0.85870215
               2  post_p_L
               3  zygomatic_W
               4  palatal_W1
               5  palatal_W2
               6  postg_foramina_W
               7  interorbital_W
               8  braincase_W
```

PART 3 : p-Values for Bartlett's Modified LR ECM Test

What this problem is.

A description of the univariate version of Bartlett's Modified Likelihood Ratio Test can be found on the following web sites: http://www.itl.nist.gov/div898/handbook/eda/section3/eda357.htm "1.3.5.7.Bartlett's Test" and http://www.itl.nist.gov/div898/handbook/eda/section3/eda3581.htm

What is computed in this package is a multivariate version, a version with mutiple dependent variables. In terms of a data matrix it can be thought of as the transpose of block independence computed with `statsBIf` above. In it, a data matrix is segregated by rows instead of columns, into two groups. A test of equality of covariances between the two groups is then performed.

The univariate version of this Bartlett's test depends on a Chi-Squared distribution. The multivariate version, in turn, depends of the multivariate version of the Chi-Squared distribution, the Wishart distribution, in particular a ratio of two Wishart distributed random variables.

The distribution of this multivariate Bartlett test is known, assuming random variable normality, as is its moment generating function. Bulter and Wood show how the Lugannani and Rice approximation of this known distribution can be used to derive power and p-value estimates otherwise unobtainable.

Let $x_1, \cdots, x_{N1}$ be a matrix with N1 rows and p columns with data sampled from a multivariate normal

distribution, $N_p(\mu_1, \Sigma_1)$. And assume there is another matrix $x_1, \cdots, x_{N2}$ a matrix with N2 rows and (the same) p columns sampled from a multivariate normal distribution with μ_2 and Σ_2. $N1$ does not have to be equal to $N2$. μ_1 does not have to equal μ_2.

The null hypothesis of equal covariance matrices (ECM) is:

$$\Sigma_1 = \Sigma_2 = \Sigma$$

A likelihood ratio test in this case is can be

$$\Lambda_{ECM} = \frac{(|A_1|)^{\frac{n1}{n}} * (|A_2|)^{\frac{n2}{n}}}{(|(A_1 + A_2)|)}$$

where $|\bullet|$ is the determinant of a matrix. A_1 and A_2 are the estimated variance covariance matrices for the two data matrices. The distribution of this statistic is a ratio of two independently distributed Wishart random variables, the noncentral version of which is determined by $n1 = N1 - 1$,$n2 = N2 - 1$,$n = n1 + n2$,p,and the estimated eigenvalues of the matrix $\Sigma_1 * \Sigma_2^{-1}$.

Example 1: Plastic Dataset, Additive Variable, statsECMf

```
    suppressPackageStartupMessages(library(Rmpfr));
    library(wmpvaer);
    #prepare data
    data(plastic)

    # select obs w low additive
    a1<-plastic[plastic$additive=="Low",];
    # create matrix w dependent vars
    low_additive<-as.matrix(a1[,1:3]);

    # select obs w high additive
    a2<-plastic[!(plastic$additive=="Low"),];
    # create matrix w dependent vars
    high_additive<-as.matrix(a2[,1:3]);

    # run statsECMf with defined matrices
    statsECMf(low_additive,high_additive);
lambdaECM   p value
0.049692494 7.3611528e-06
```

In all wmpvaer datasets the value of Λ_{ECM} estimated is small as is the estimated p-value. The test rejects the null hypothesis of equality of covariance matrices for all datasets that were examined.

If Σ_1 is equal to Σ_2 and both are invertible matrices then $\Sigma_1 * \Sigma_2^{-1} = I$,an identity matrix, with all of its eigenvalues equal to 1. In the plastic additive variable example above the estimated $\Sigma_1 * \Sigma_2^{-1}$ is:

Table 1: with Eigenvalues

	tear	gloss	opacity
tear	0.9203183	-1.526514	-0.1333428
gloss	-1.1579138	2.457798	0.2470928
opacity	-1.1065895	4.094780	0.7460240

additive_eigs	3.621349	0.4210647	0.0817264

This matrix is far from an identity matrix and the eigenvalues are not a vector of 1s. Unless this matrix is some reasonable approximation of an identity matrix, Bartlett's modified LR ECM test is likely to reject the null hypothesis of equal covariance matrices.

Computational issue.

For both the block independence and equivalence of covariance matrices program (`statsBIf` and `statsECMf` respectively) bisection search alone would not converge with an accurate answer. In both sets of programs, bisection search is augmented by a preliminary scan of possible solutions. For `statsBIf` this is done at fixed intervals. For `statsECMf` the solutions are selected using a random sample.

For `statsECMf`, if the program is executed repeatedly on the same dataset, slightly different p-values can result.

PART 4: Butler/Wood Saddlepoint Solution, Sample Size Estimation, Balanced 1-Way MANOVA Design

Method Outline

In a preliminary version of their 2005 paper, Butler and Wood outlined the following method for estimating the number of replications needed for a 1-way MANOVA design.

Define $G_0(x) := Pr(log(Wilks) <= x|\Omega = 0p, p)$.

Recall that the p-value (type 2 error) is defined to be $Pr(x <= log(Wilks)|\Omega_{observed})$. Conversely the power value is $Pr(x > log(Wilks)|\Omega_{observed})$ or 1-(type 2 error). So $Pr(log(Wilks) <= x|\Omega = 0p, p)$ (assuming $\Omega = 0p, p$) is the probability of a type 1 error (or probability of a false positive, incorrectly rejecting a null hypothesis when it should not be rejected). The Butler/Wood equations used to estimate this function are the same as those used to estimate a p-value when Ω is an observed value instead of an assumed null value $\Omega = 0p, p$. In a study design $G_0(x) := Pr(log(Wilks) <= x|\Omega = 0p, p)$ is often referred to as alpha (α).

Call the approximation of $G_0(x)$ $\widehat{G_0}(x)$. Then

1. From this find $x_0 := \widehat{G_0^{-1}}(.05)$, where x_0 is the Wilks value that gives a type 1 error probability of .05.
2. Make the assumption that you know what Ω_1 (the vector of eigenvalues) is for your study. This is guessed at from the result of past studies or it can simply be set hypothetically.
3. The power function for this Ω_1 is $G_{\Omega_1}(x, J) := Pr(x > log(Wilks)|\Omega = J * \Omega_1)$ (where J is the number of replications to be used in the study). And its estimate is $\widehat{G_{\Omega_1}}(x, J)$.
4. Plot $\widehat{G_{\Omega_1}}(x_0, J)$ vs J=2,3,4..... From the plot estimate what J value results in $\widehat{G_{\Omega_1}}(x_0, J) = .9$. Or computationally find J such that $\widehat{G_{\Omega_1}}(x_0, J) = .9$. The value of J resulting will give you an estimate of the number of replications needed so that your Wilks statistic will have an alpha level of .05 and an a-priori power value of .9. This is your guess anyway before the study is conducted and the data is collected.

This method is not generally practical with the R code in this package (as Butler and Wood described the method). The power function and alpha estimate also require that the error and model degrees of freedom be specified. As the number of error,model degrees of freedom and dimension of Ω increases the question of

just where exactly $x_0 := \widehat{G_0^{-1}}(.05)$ becomes unanswerable as is the question of where $\widehat{G_{\Omega_1}}(x_0, J) = .9$. This is a limitation of this R code.

If the question is "$x_0 :\approx \widehat{G_0^{-1}}(.05)$, where x_0 is the Wilks value that gives a type 1 error probability close to the x_0 that equals .05." and where "$\widehat{G_{\Omega_1}}(x_0, J) >= .9$" instead of "= .9" then the problem is computable (within limits) with the code created.

These computational limits are: 1. No less than 2 and no more than 7 dependent variables; and 2. The program currently can handle no more than 12 levels in the independent variable used in the 1-way MANOVA analysis.

Samples of Data and R Coding : `oneway_mss`

The dataset (stored as "chem_exp") ,used here to demonstrate the R function 'oneway_mss', is of unknown origin. It appears from a chemistry experiment.

There are 45 observations and 8 columns. The columns are titled (in order) "H2S, P,K,Ca,Mg,Na,Mn,Zn". It is assumed "H2S" is the independent variable, and is the level of hydrogen sulfide applied to some unknown substance. The 7 other columns appear to be dependent variables corresponding to measurements of Phosphorus,Potassium,Calcium,Magnesium,Sodium,Manganese and Zinc. "H2S" has 11 distinct values (all the other variables have 45 for each variable). These levels and the number of observations taken at each level are:

H2S levels	-640	-630	-620	-610	-600	-590	-580	-570	-560	-550	-540
obs per level	5	6	6	8	3	2	2	5	3	2	2

If you perform a 1-Way MANOVA analysis using the R lm( ) function, using H2S as a factor, and derive the (E-inverse*H) eigenvalues, you get the following vector: `H2S_eigs<-c(1.317766, 1.004767, 0.4139226, 0.2641087, 0.1737016, 0.08137942, 0.01661141)`. This can be used with `oneway_mss` as follows:

Example 1: chem_exp dataset, factor H2S, R function `oneway_mss`

```
    suppressPackageStartupMessages(library(Rmpfr));
    library(wmpvaer);
    H2S_eigs<-c(1.317766, 1.004767, 0.4139226, 0.2641087, 0.1737016, 0.08137942, 0.01661141)
    H2S_n_level<-11;
    oneway_mss(H2S_n_level,H2S_eigs);

1-Way MANOVA Balanced A-priori Study Design
---------------------------
eigs(E-inv*H)= 1.317766 1.004767 0.4139226 0.2641087 0.1737016 0.08137942 0.01661141
levels= 11 ,dependent vars= 7
alpha approx.=.05 ,power>=.9
replications needed= 2
total sample size= 22
```

So if this study was repeated or the eigenvalues derived were used in the design of a new study , the requirement (in both cases) that alpha=.05 approximately and power was >=.9 could be met with 22 observations or two replications for each level.

In some cases, the estimate of the power does not exceed .9 even after 20 replications. In that case the program is stopped when replications appear to exceed 20.

NOTE: BISECTION SEARCH FAILURE

The programs employ bisection search at some stages. Bisection search is an elementary search method. Bisection search can fail. When the bisection search for an estimation of power fails (does not converge with enough accuracy), an error statement (with the text "No Answer") appears.

NOTE: PARALLEL COMPUTATION

This version of wmpvaerf,mlandrf was rewritten to take advantage of the R package "doParallel". The program statsBIncatf employed that R package from the beginning. These versions employ only one core if only one or two cores are available. Otherwise, the number of computer cores used is the minimum of "the number of independent variables in the model in question" or "the maximum number of available cores minus one" in the case of wmpvaerf and mlandrf. In the case of statsBIncatf, only the last condition applies.