THE UNIVERSITY OF MINNESOTA

Statistics 5401 September 26, 2005

A Simple Approach to Simulation Using MacAnova

There are many situations when you can use simulation to get an approximate P-value of a test, or to find approximate critical values of a test statistic.

To do this in MacAnova you need to be able to accomplish two tasks.

(1) Generate a matrix `of` artificial data which (a) matches the assumptions of your analysis and (b) for which the null hypothesis is true.

(2) Compute the value `testval` of the test statistic from the data.

Assume `generate()` represents commands that generate `y` and `compute(y)` represents commands that computes `testval` from `y`. Then, if `M` is the number of simulations you use, a general way to do the simulations in MacAnova is as follows:

```
Cmd> values <- rep(0,M) # set up vector for simulated values
Cmd> for(i,1,M){   # loop M times with i going from 1 to M
       y <- generate() # generate data matrix
       testval <- compute(y) # compute test statistic
       values[i] <- testval;; # save test statistic
     }
```

You can now use the results in `values` to estimate a P-value or find a critical value.

Suppose, your actual data is matrix `y_observed` and the observed value of the test statistic is `testval_obs` computed by

```
Cmd> testval_obs <- compute(y_observed)
```

After the simulation, assuming the test is "reject for large values of `testval`", you can estimate the P-value by

```
Cmd> p_value <- sum(values >= testval_obs)/M
```

Here `sum(values >= testval_obs)` counts how many simulated values are at least as large as the observed value. Dividing by M gives the relative frequency of such values and this estimates P-value = P(`testval` ≥ `testval_obs` | H_0).

The critical value corresponding, say, to $\alpha = .05$, is the $100(1 - \alpha)$ percent point of the test statistic. This can be estimated as the $100(1 - \alpha)$ percent point of the sample of simulated values. For an upper 5% point you might do the following:

```
Cmd> values <- sort(values) # put test values in increasing order
Cmd> J <- round(.95*M) # index of approximate 95% point
Cmd> critval <- values[J]
```

Sometimes the most difficult part of this process is knowing how to generate data which satisfy the null hypothesis. Here you may need some mathematical results to succeed.

For example, the null distributions of most tests used in ANOVA, including F-tests and t-tests, do not depend on the value of the variance σ^2 so you can use any convenient value such as $\sigma^2 = 1$. Moreover, as long as H_0 is true, the distributions don't depend on any mean values,

so you can use $\mu = 0$.

For many multivariate tests assuming normal data or normal residuals with constant variance matrix $\mathbf{\Sigma}$, the null distribution does not depend on $\mathbf{\Sigma}$ so you can use any convenient $\mathbf{\Sigma}$, say $\mathbf{\Sigma} = \mathbf{I}_p$, that is, the p responses are independent with variance 1. Similarly, often the null distributions don't depend on mean values so you can use $\boldsymbol{\mu} = \mathbf{0}$. Note, however, the *joint* distribution of univariate test statistics such as F-statistics do depend on $\mathbf{\Sigma}$, so you can't use $\mathbf{\Sigma} = \mathbf{I}_p$, even though the *marginal* distributions do not depend on $\mathbf{\Sigma}$.

In MacAnova, you can generate a data matrix `x` containing a random sample of n $N_p(0, \mathbf{I}_p)$ vectors by `x <- matrix(rnorm(n*p), n)`.

Here is a MANOVA example based on the analysis of data on crude oil in Table 11.7, 9. 661. See problem 11.30, p. 660 J&W for a description of the data.

```
Cmd> data <- read("","t11_07") # read from JWData5.txt
T11_07    56    6 format
) Data from Table 11.7 p. 661 in
) Applied Mulivariate Statistical Analysis, 5th Edition
) by Richard A. Johnson and Dean W. Wichern, Prentice Hall, 2002
) These data were edited from file T11-7.DAT on disk from book
) Group identification was moved from last column to first and
) made numeric
) Crude oil data
) Col. 1: Zone (1 = Wilhelm, 2 = sub-Mulinia, 3 = Upper
) Col. 2: X1 = vanadium (percent ash)
) Col. 3: X2 = iron (percent ash)
) Col. 4: X3 = beryllium (percent ash)
) Col. 5: X4 = saturated hydrocarbons (percent area)
) Col. 6: X5 = aromatic hydrocarbons (percent area)
Read from file "TP1:Stat5401:Data:JWData5.txt"

Cmd> zone <- factor(data[,1]); y <- data[,-1]
```

When you use `byvar:T` on a `manova()` command, the output is in the form of p univariate analyses of variance, one for each response. However, the usual side-effect variables are computed.

```
Cmd> manova("y = zone", byvar:T, fstat:T)
Model used is y = zone
WARNING: summaries are sequential
                          Variable 1
                DF          SS          MS           F     P-value
CONSTANT         1        2139        2139   604.38757     < 1e-08
zone             2      135.67      67.837    19.16745  5.4505e-07
ERROR1          53      187.58      3.5392
                          Variable 2
                DF          SS          MS           F     P-value
CONSTANT         1       40965       40965   514.34216     < 1e-08
zone             2      3186.7      1593.3    20.00566   3.366e-07
ERROR1          53      4221.2      79.644
                          Variable 3
                DF          SS          MS           F     P-value
CONSTANT         1      6.5281      6.5281    78.00184     < 1e-08
zone             2     0.98442     0.49221     5.88122   0.0049345
ERROR1          53      4.4357    0.083692
                          Variable 4
                DF          SS          MS           F     P-value
CONSTANT         1      1572.5      1572.5  1461.12200     < 1e-08
zone             2      48.803      24.402    22.67323  7.6772e-08
ERROR1          53       57.04      1.0762
                          Variable 5
                DF          SS          MS           F     P-value
CONSTANT         1      2317.9      2317.9   363.43106     < 1e-08
zone             2      209.29      104.65    16.40805  2.8427e-06
ERROR1          53      338.02      6.3778

Cmd> list(SS,DF) # side-effect variables have been computed
DF              REAL   3
SS              REAL   3     5     5

Cmd> h <- matrix(SS[2,,]); e <- matrix(SS[3,,])# hyp & error matrices

Cmd> fh <- DF[2]; fe <- DF[3]; vector(fh,fe)# hyp & error d.f.
        zone      ERROR1
           2          53

Cmd> eigvals <- releigenvals(h,e); eigvals #obs. relative eigenvalues
(1)      4.1784     0.66601  2.0476e-16  3.7444e-18 -2.9572e-17

Cmd> N <- nrows(y); p <- ncols(y); vector(N,p)
(1)          56           5

Cmd> m1 <- fe - (p - fh + 1)/2; m2 <- fe - p - 1; m3 <- fh + fe

Cmd> vector(m1, m2, m3) # multipliers for test statistics
(1)          51          47          55

Cmd> wilks_obs <- m1*sum(log(1 + eigvals)) # observed Wilks'

Cmd> hot_obs <- m2*sum(eigvals) # observed Hotelling's

Cmd> pillai_obs <- m3*sum(eigvals/(1 + eigvals)) # observed Pillai's

Cmd> roy_obs <- eigvals[1] # observed Roy's (maximum root)

Cmd> vector(wilks_obs,hot_obs,pillai_obs,roy_obs)
(1)       109.9      227.69      66.366      4.1784
```

```
Cmd> wilks_obs <- m1*sum(log(1 + eigvals)) # observed Wilks'
Cmd> hot_obs <- m2*sum(eigvals) # observed Hotelling's
Cmd> pillai_obs <- m3*sum(eigvals/(1 + eigvals)) # observed Pillai's
Cmd> roy_obs <- eigvals[1] # observed Roy's (maximum root)
Cmd> vector(wilks_obs,hot_obs,pillai_obs,roy_obs)
(1)       109.9      227.69      66.366      4.1784
```

`wilks_obs`, `hot_obs`, `pillai_obs`, and `roy_obs` are the observed values Wilks's Λ, Hotelling's trace statistic, Pillai's trace statistic, and Roy's maximum root statistic, all standard MANOVA test statistics of the hypothesis H_0: $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = \boldsymbol{\mu}_3 = \boldsymbol{\mu}$.

Now do 5000 simulations with H_0 true. Since the null distributions don't depend on $\boldsymbol{\mu}$ or $\boldsymbol{\Sigma}$, I use $\boldsymbol{\mu} = \mathbf{0}$ and $\boldsymbol{\Sigma} = \mathbf{I}_5$. This means that each data matrix can consist of N×p independent standard normals.

```
Cmd> M <- 5000;hot <- wilks <- pillai <- roy <- rep(0,M)#for values
Cmd> for(i,1,M){
       # do the generate() step
       ytmp <- matrix(rnorm(N*p),N) # simulated data matrix
       # do the compute() step, but compute 4 statistics at once
       manova("ytmp = zone", silent:T) # silently do MANOVA
       eigtmp <- releigenvals(SS[2,,],SS[3,,]) #relative eigenvalues
       wilks[i] <- m1*sum(log(1 + eigtmp))
       hot[i] <-  m2*sum(eigtmp)
       pillai[i] <- m3*sum(eigtmp/(1 + eigtmp))
       roy[i] <- eigtmp[1]
       ;;
   }
```

Vectors `wilks`, `hot`, `pillai` and `roy` contain samples of the 4 statistics. They need to be put in increasing order before finding 10%, 5%, 2.5% and 1% critical values.

```
Cmd> wilks <- sort(wilks); hot <- sort(hot)
Cmd> pillai <- sort(pillai); roy <- sort(roy)
Cmd> alpha <- vector(.1,.05,.025,.01) # 10%, 5%, 2.5%, 1%
Cmd> J <- round((1 - alpha)*M); J # indices of probability points
(1)        4500        4750        4875        4950
Cmd> wilks[J] # critical values for Wilk's test
(1)      15.897      18.191      20.181       22.25
Cmd> hot[J] # critical values for Hotelling's test
(1)      16.242       18.93       21.28      24.002
Cmd> pillai[J] # critical values for Pillai's test
(1)      15.481      17.521      19.183      20.738
```

In large samples, the null distribution of each of these is approximately χ^2 on $f_h p$ d.f. Here p = 5, $f_h = 2$ and the asymptotic χ^2_{10} critical values are computed as follows:

```
Cmd> invchi(alpha,p*fh,upper:T) # chi-squared critical values
(1)      15.987      18.307      20.483      23.209
```

Estimate the actual α's if you use these large sample critical values with the tests.

```
Cmd> sum(wilks > invchi(alpha,p*fh,upper:T)')/M #estimated alphas
(1,1)      0.0978      0.0492       0.022      0.0072
Cmd> sum(hot > invchi(alpha,p*fh),upper:T)')/M #estimated alphas
(1,1)      0.1064      0.0594      0.0324      0.0128
Cmd> sum(pillai > invchi(alpha,p*fh),upper:T)')/M #estimated alphas
(1,1)      0.0856      0.0354       0.012      0.0028
```

All the α's appear to be in the right ballpark, except possibly for Pillai's statistic.
Of course, these are just estimates. Using standard binomial theory, here are 95% margins of error for them.

```
Cmd> 1.96*sqrt(alpha*(1 - alpha)/M) # 95% margins of error
(1)   0.0083156   0.0060411   0.0043276    0.002758
```

In fact, only the α's for the Wilks' statistic are consistently not significantly different from the intended α's.

Here are estimated critical values for Roy's maximum root test, both in terms of $\hat{\lambda}_1$ and $\hat{\theta}_1 = \hat{\lambda}_1/(1+\hat{\lambda}_1)$.

```
Cmd> roy[J] # maximum root critical values
(1)     0.27729     0.32476     0.37476     0.43584
Cmd> (roy/(1 + roy))[J] # critical values for theta
(1)     0.21709     0.24515      0.2726     0.30354
Cmd> vector(min(p,fh), (abs(fh - p) - 1)/2, (fe - p - 1)/2) # s, m, n
(1)           2           1        23.5
```

These are the values you use with charts or tables of the null distribution.

Since the null hypothesis is so strongly rejected in the ANOVA F-tests, we should expect the P-values to be small. In fact, for each statistic, the observed value is greater than any simulated value so the P-values are all estimated to be 0.

```
Cmd> sum(wilks >= wilks_obs)/M # P-value for Wilks'
(1)           0
Cmd> sum(hot >= hot_obs)/M # P-value for Hotelling's
(1)           0
Cmd> sum(pillai >= pillai_obs)/M # P-value for Pillai's
(1)           0
Cmd> sum(roy >= roy_obs)/M # P-value for Roy's
(1)           0
```

Thus, even in the absence of tables or charts, you could conclude with high confidence that the null hypothesis of equal means was false.