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

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

You can now use the results in \( \text{values} \) to estimate a P-value or find a critical value.

Suppose, your actual data is matrix \( y_{\text{observed}} \) and the observed value of the test statistic is \( \text{testval}_{\text{obs}} \) computed by

\[
\text{Cmd>} \quad \text{testval}_{\text{obs}} \leftarrow \text{compute}(y_{\text{observed}})
\]

After the simulation, assuming the test is "reject for large values of \( \text{testval} \)”, you can estimate the P-value by

\[
\text{Cmd>} \quad \text{p-value} \leftarrow \frac{\text{sum(values} \geq \text{testval}_{\text{obs}})}{M}
\]

Here \( \text{sum(values} \geq \text{testval}_{\text{obs}}) \) 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 = \( \text{P} (\text{testval} \geq \text{testval}_{\text{obs}} | H_0) \).

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:

\[
\text{Cmd>} \quad \text{values} \leftarrow \text{sort(values)} \quad \text{# put test values in increasing order}
\]
\[
\text{Cmd>} \quad J \leftarrow \text{round(.95*M)} \quad \text{# index of approximate 95% point}
\]
\[
\text{Cmd>} \quad \text{critval} \leftarrow \text{values}[J]
\]

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 \( \sigma^2 \) 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 \( \Sigma \), the null distribution does not depend on \( \Sigma \) so you can use any convenient \( \Sigma \), say \( \Sigma = I_p \), 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 \( \mu = 0 \). Note, however, the joint distribution of univariate test statistics such as F-statistics do depend on \( \Sigma \), so you can’t use \( \Sigma = I_p \), even though the marginal distributions do not depend on \( \Sigma \).

In MacAnova, you can generate a data matrix \( x \) containing a random sample of \( n N_p(0, I_p) \) 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, p. 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
)  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

<table>
<thead>
<tr>
<th></th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONSTANT</td>
<td>1</td>
<td>2139</td>
<td>2139</td>
<td>604.38757</td>
<td>&lt; 1e-08</td>
</tr>
<tr>
<td>zone</td>
<td>2</td>
<td>135.67</td>
<td>67.837</td>
<td>19.16745</td>
<td>5.4505e-07</td>
</tr>
</tbody>
</table>

Variable 2

<table>
<thead>
<tr>
<th></th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONSTANT</td>
<td>1</td>
<td>40965</td>
<td>40965</td>
<td>514.34216</td>
<td>&lt; 1e-08</td>
</tr>
<tr>
<td>zone</td>
<td>2</td>
<td>3186.7</td>
<td>1593.3</td>
<td>20.00566</td>
<td>3.366e-07</td>
</tr>
</tbody>
</table>

Variable 3

<table>
<thead>
<tr>
<th></th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONSTANT</td>
<td>1</td>
<td>6.5281</td>
<td>6.5281</td>
<td>78.00184</td>
<td>&lt; 1e-08</td>
</tr>
<tr>
<td>zone</td>
<td>2</td>
<td>0.98442</td>
<td>0.49221</td>
<td>5.88122</td>
<td>0.0049345</td>
</tr>
</tbody>
</table>

Variable 4

<table>
<thead>
<tr>
<th></th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONSTANT</td>
<td>1</td>
<td>1572.5</td>
<td>1572.5</td>
<td>1461.12200</td>
<td>&lt; 1e-08</td>
</tr>
<tr>
<td>zone</td>
<td>2</td>
<td>48.803</td>
<td>24.402</td>
<td>22.67323</td>
<td>7.672e-08</td>
</tr>
</tbody>
</table>

Variable 5

<table>
<thead>
<tr>
<th></th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONSTANT</td>
<td>1</td>
<td>2317.9</td>
<td>2317.9</td>
<td>363.43106</td>
<td>&lt; 1e-08</td>
</tr>
<tr>
<td>zone</td>
<td>2</td>
<td>209.29</td>
<td>104.65</td>
<td>16.40805</td>
<td>2.8427e-06</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>DF</th>
<th>REAL</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS</td>
<td>REAL</td>
<td>3</td>
<td>5</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4.1784</td>
</tr>
<tr>
<td></td>
<td>0.66601</td>
</tr>
<tr>
<td></td>
<td>2.0476e-16</td>
</tr>
<tr>
<td></td>
<td>3.7444e-18</td>
</tr>
<tr>
<td></td>
<td>-2.9572e-17</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>5</td>
</tr>
</tbody>
</table>

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

Cmd> vector(m1, m2, m3) # multipliers for test statistics

<table>
<thead>
<tr>
<th></th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>55</td>
</tr>
</tbody>
</table>

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)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>109.9</td>
</tr>
<tr>
<td></td>
<td>227.69</td>
</tr>
<tr>
<td></td>
<td>66.366</td>
</tr>
<tr>
<td></td>
<td>4.1784</td>
</tr>
</tbody>
</table>
A Simple Approach to Simulation Using MacAnova

\begin{verbatim}
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 H₀: μ₁ = μ₂ = μ₃ = μ.

Now do 5000 simulations with H₀ true. Since the null distributions don't depend on μ or Σ, I use μ = 0 and Σ = I₅. 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 χ² on fₕp d.f. Here p = 5, fₕ = 2 and the asymptotic χ₁₀² 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
\end{verbatim}
Estimate the actual $\alpha$’s if you use these large sample critical values with the tests.

\[
\begin{align*}
\text{Cmd}> & \quad \text{sum}(\text{wilks} > \text{invchi}(\alpha, p*fh, \text{upper:T}'))/M \ # \text{estimated alphas} \\
& (1,1) \quad 0.0978 \quad 0.0492 \quad 0.022 \quad 0.0072 \\
\text{Cmd}> & \quad \text{sum}(\text{hot} > \text{invchi}(\alpha, p*fh), \text{upper:T}'))/M \ # \text{estimated alphas} \\
& (1,1) \quad 0.1064 \quad 0.0594 \quad 0.0324 \quad 0.0128 \\
\text{Cmd}> & \quad \text{sum}(\text{pillai} > \text{invchi}(\alpha, p*fh), \text{upper:T}'))/M \ # \text{estimated alphas} \\
& (1,1) \quad 0.0856 \quad 0.0354 \quad 0.012 \quad 0.0028
\end{align*}
\]

All the $\alpha$’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.

\[
\begin{align*}
\text{Cmd}> & \quad 1.96*\sqrt{\alpha*(1 - \alpha)/M} \ # \text{95% margins of error} \\
& (1) \quad 0.0083156 \quad 0.0060411 \quad 0.0043276 \quad 0.002758
\end{align*}
\]

In fact, only the $\alpha$’s for the Wilks’ statistic are consistently not significantly different from the intended $\alpha$’s.

Here are estimated critical values for Roy’s maximum root test, both in terms of $\hat{\lambda}_i$ and

\[
\hat{\theta}_i = \hat{\lambda}_i/(1 + \hat{\lambda}_i).
\]

\[
\begin{align*}
\text{Cmd}> & \quad \text{roy}[J] \ # \text{maximum root critical values} \\
& (1) \quad 0.27729 \quad 0.32476 \quad 0.37476 \quad 0.43584 \\
\text{Cmd}> & \quad (\text{roy}/(1 + \text{roy}))[J] \ # \text{critical values for theta} \\
& (1) \quad 0.21709 \quad 0.24515 \quad 0.2726 \quad 0.30354 \\
\text{Cmd}> & \quad \text{vector}(\min(p,fh), (abs(fh - p) - 1)/2, (fe - p - 1)/2) \ # s, m, n \\
& (1) \quad 2 \quad 1 \quad 23.5
\end{align*}
\]

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.

\[
\begin{align*}
\text{Cmd}> & \quad \text{sum}(\text{wilks} >= \text{wilks}_\text{obs})/M \ # \text{P-value for Wilks’} \\
& (1) \quad 0 \\
\text{Cmd}> & \quad \text{sum}(\text{hot} >= \text{hot}_\text{obs})/M \ # \text{P-value for Hotelling’s} \\
& (1) \quad 0 \\
\text{Cmd}> & \quad \text{sum}(\text{pillai} >= \text{pillai}_\text{obs})/M \ # \text{P-value for Pillai’s} \\
& (1) \quad 0 \\
\text{Cmd}> & \quad \text{sum}(\text{roy} >= \text{roy}_\text{obs})/M \ # \text{P-value for Roy’s} \\
& (1) \quad 0
\end{align*}
\]

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