```
*==========================================================================================
* test rCategorical by generating 1000 values & checking proportions
clear
program drop _all

program rCat
        syntax newvarlist(max=1) , N(integer) P(namelist)

        if `n' > _N qui set obs `n'
        qui gen `varlist' = .
        mata: P = st_matrix("`p'")
        forvalues i=1/`n' {
            mata: H = rCategorical(P)
                mata: st_numscalar("r(h)",H)
                qui replace `varlist' = r(h) in `i'
        }
end

matrix PR = (0.2,0.4,0.1,0.3)
rCat x , n(1000) p(PR)
tabu x
*==========================================================================================
* test rDirichlet by generating 1000 values & checking proportions match alpha
clear
program drop _all

program rDir
        syntax newvarlist , N(integer) Alpha(namelist)

        if `n' > _N qui set obs `n'
        foreach v of newlist `varlist' {
           qui gen `v' = .
    }
        mata: A = st_matrix("`alpha'")
        forvalues i=1/`n' {
            mata: P = rDirichlet(A)
                mata: st_matrix("pm",P)
                local k = 0
                foreach v of local varlist {
                   local ++k
                qui replace `v' = pm[`k',1] in `i'
                }
    }
end

matrix A = (10,10,5,5)
rDir p1 p2 p3 p4 , n(1000) alpha(A)
su p*

*==========================================================================================
* test rMNormal by generating 1000 (similar to drawnorm)
clear
program drop _all

program rMN
        syntax newvarlist , N(integer) Mean(namelist) Var(namelist)

        if `n' > _N qui set obs `n'
        foreach v of newlist `varlist' {
           qui gen `v' = .
    }
        mata: M = st_matrix("`mean'")
        mata: V = st_matrix("`var'")
        mata: L = cholesky(V)
        forvalues i=1/`n' {
            mata: X = rMNormal(M,L)
                mata: st_matrix("x",X)
                local k = 0
                foreach v of local varlist {
                   local ++k
                qui replace `v' = x[`k',1] in `i'
                }
    }
end

matrix M = (3 \ 4 \ 5)
matrix V = (2,1,-1\1,4,0\-1,0,2)
```

```
rMN x1 x2 x3 , n(1000) mean(M) var(V)
matrix accum S = x1 x2 x3 , nocon dev mean(mn)
matrix list mn
matrix S = S/1000
matrix list S

*==========================================================================================
* test rWishart by having a large scalar parameter so that the variance matrix is
* accurately approximated
clear
program drop _all

program rWi
        syntax name ,  DF(integer) R(namelist)
        mata: R = st_matrix("`r'")
        mata: L = cholesky(invsym(R))
        scalar DF = `df'
    mata: K = st_numscalar("DF")
        mata: X = rWishart(L,K)
        mata: st_matrix("`namelist'",X)
end

matrix V = (2,1,-1\1,4,0\-1,0,2)
matrix R = 500*V
rWi W , df(500) r(R)
matrix list W
matrix iW = syminv(W)
matrix list iW
```