1 Short Help on random and randn

You can use the *random* command to generate random vectors and matricies. The following illustrates the syntax of the *random* command.

```
R=random(’ProbabilityDistribution’,Parameter1,..,ParameterN,#rows,#cols);
```

For *Gaussian* or *Normal*, the distribution is of type ’norm’, *Parameter1* is the mean, and *Parameter2* is the standard deviation. For this type of distribution, only these parameters are required. Remember that the standard deviation is the square root of the variance. A listing of the supported distributions for *random* is contained in Matlab’s help file, also included in the section below.

Example: Suppose you want to generate a signal vector of Gaussian noise. To generate a row vector of length 10, containing Gaussian distributed numbers with mean 5 and variance 2, you would type

```
R=random(’norm’,5,sqrt(2),1,10);
```

The Matlab command *randn* generates samples of a Gaussian distributed random variable with mean 0 and variance 1. To obtain a mean other than zero, just add or subtract a constant from the generated vector. To obtain a variance other than one, multiply the generated vector by the standard deviation (square root of the variance).

Example: To generate a length 10 row vector of Gaussian distributed numbers with mean 5 and variance 2, you would type

```
R=sqrt(2).*randn(1,10) + 5;
```

Note that *randn* executes significanlty faster than *random*, so this would be the preferred function in longer simulations.

2 Matlab Help on random

```
 RANDOM Generate random arrays from a specified distribution.
    R = RANDOM(NAME,A) returns an array of random numbers chosen from the
    one-parameter probability distribution specified by NAME with parameter
    values A.

    R = RANDOM(NAME,A,B) or R = RANDOM(NAME,A,B,C) returns an array of random
    numbers chosen from a two- or three-parameter probability distribution
    with parameter values A, B (and C).

    The size of R is the common size of the input arguments.  A scalar input
```

```
functions as a constant matrix of the same size as the other inputs.

R = RANDOM(NAME,A,M,N,...) or R = RANDOM(NAME,A,[M,N,...]) returns an
M-by-N-by-... array of random numbers for a one-parameter distribution.
Similarly, R = RANDOM(NAME,A,B,M,N,...) or R = RANDOM(NAME,A,B,[M,N,...]),
and R = RANDOM(NAME,A,B,C,M,N,...) or R = RANDOM(NAME,A,B,C,[M,N,...]),
return an M-by-N-by-... array of random numbers for a two- or
three-parameter distribution.


NAME can be:

   ’beta’  or ’Beta’,
   ’bino’  or ’Binomial’,
   ’chi2’  or ’Chisquare’,
   ’exp’   or ’Exponential’,
   ’ev’    or ’Extreme Value’,
   ’f’     or ’F’,
   ’gam’   or ’Gamma’,
   ’gev’   or ’Generalized Extreme Value’,
   ’gp’    or ’Generalized Pareto’,
   ’geo’   or ’Geometric’,
   ’hyge’  or ’Hypergeometric’,
   ’logn’  or ’Lognormal’,
   ’nbin’  or ’Negative Binomial’,
   ’ncf’   or ’Noncentral F’,
   ’nct’   or ’Noncentral t’,
   ’ncx2’  or ’Noncentral Chi-square’,
   ’norm’  or ’Normal’,
   ’poiss’ or ’Poisson’,
   ’rayl’  or ’Rayleigh’,
   ’t’     or ’T’,
   ’unif’  or ’Uniform’,
   ’unid’  or ’Discrete Uniform’,
   ’wbl’   or ’Weibull’.

Partial matches are allowed and case is ignored.

RANDOM calls many specialized routines that do the calculations.

See also cdf, icdf, mle, pdf.
```

3 Matlab Help on randn

```
 RANDN  Normally distributed random numbers.
    R = RANDN(N) returns an N-by-N matrix containing pseudo-random values
    drawn from a normal distribution with mean zero and standard deviation
    one.  RANDN(M,N) or RANDN([M,N]) returns an M-by-N matrix.  RANDN(M,N,P,...)
    or RANDN([M,N,P,...]) returns an M-by-N-by-P-by-... array.  RANDN with
    no arguments returns a scalar.  RANDN(SIZE(A)) returns an array the
    same size as A.

    You can use one of two generator algorithms, as follows:

    RANDN(METHOD,S) causes RANDN to use the generator determined by METHOD, and
    initializes the state of that generator.  S is a scalar integer value from
    0 to 2^32-1, or the output of RANDN(METHOD).  METHOD is one of the
    following strings:

       ’state’ - Use Marsaglia’s Ziggurat algorithm, the default in MATLAB
                 Versions 5 and later.  The period is approximately 2^64.

       ’seed’  - Use the polar algorithm, the default in MATLAB Version 4.
                 The period is approximately (2^31-1)*(pi/8).

    RANDN(METHOD) returns the current internal state of the generator
    determined by METHOD.  However, it does not switch generators.

    The sequence of numbers produced by RANDN is determined by the internal
    state of the generator.  Setting the generator to the same fixed state
    allows computations to be repeated.  Setting the generator to different
    states leads to unique computations, however, it does not improve any
    statistical properties.  Since MATLAB resets the state at start-up, RANDN
    will generate the same sequence of numbers in each session unless the
    state is changed.

    Note: The size inputs M, N, and P... should be nonnegative integers.
    Negative integers are treated as 0.

     Examples:

       Return RANDN to its default initial state.
          randn(’state’,0)

       Initialize RANDN to a different state each time.
          randn(’state’,sum(100*clock))
```

```
        Save the current state, generate 100 values, reset the state,
        and repeat the sequence.
           s = randn('state');
           r1 = randn(100);
           randn('state',s);
           r2 = randn(100); % contains exactly the same values as r1

        Generate normal values with mean 1 and standard deviation 2.
           r = 1 + 2.*randn(100,1);

    See also rand, sprand, sprandn, randperm.
```