RANDOM

RANDOM (CAUCHY, DF=*scalar*, DRAW=*series or matrix*, EDF=*series*, EXPON, GAMMA, GEN=1 or 2, LAMBDA=*scalar*, LAPLACE, MEAN=*scalar*, NEGBIN, POISSON, SEEDIN=*scalar*, SEEDOUT=*scalar*, STDEV=*scalar*, T, UNIFORM, VCOV=*vcov matrix*, VMEAN=*mean vector*)
series or *matrix* or *list of series* ;

Function:

RANDOM creates pseudo-random variables. It can create random variables which follow the normal, uniform, Poisson, Negative Binomial, Laplace, t, Cauchy, exponential, gamma, or an empirical distribution. The user may specify (optionally) parameters of the distribution.

Usage:

RANDOM with no options causes a normal random variable with mean zero and variance one to be generated and stored as a series (under control of the current SMPL). If you want a non-standardized random variable, include the MEAN and STDEVoptions.

Other options cause the random variable generated to follow the Poisson, negative binomial, uniform, t, Cauchy, exponential, gamma, Laplace (double exponential), or general empirical distributions. See the Examples Section to learn how to obtain random variables from other distributions.

TSP "randomizes" the seed to start every run based on the current time, so you need to specify a fixed seed if you want to reproduce results from run to run. To change the starting seed for any given run or to save it for future use, use the SEEDIN and SEEDOUT options.

If multivariate normal random deviates are desired, the VCOV option is required. The dimension of this (symmetric or diagonal) matrix is the number of series to create. If only one argument is supplied before the semicolon, the series are stored in a matrix with this name. The VMEAN option (not the scalar MEAN option) is used if a non-zero mean vector is desired.

To create a random variable from an empirical distribution function, a series (usually a set of residuals) generated by the distribution function must be supplied. This feature is useful for computing bootstrap standard errors. A set of residuals generated by the model is used as input to the DRAW option and a new series with the same distribution as the old one is obtained by drawing observations from a discrete distribution with probability mass equal to one divided by the number of observations placed on each observed value of the residuals. This new sample of residuals may then be used in further computations to obtain estimates of functions of these random variables.

Options:

CAUCHY/**NOCAUCHY** specifies that the random number generated is to follow the Cauchy distribution
($f(x) = 1/[\pi(1+x^2)]$).

DF = the degrees of freedom for student's t distribution (see the t option).

DRAW = the name of a series which will be sampled with probability 1/@NOB to generate the random numbers. This series does not have to be sorted in any order. If DRAW= a matrix, a multivariate set of random numbers is drawn. The number of random variables is equal to the number of columns in the matrix. Note that the matrix from which you are drawing does not have to have the number of rows equal to the number of observations. This is very useful for simulation or bootstrap standard errors.

EDF = Empirical Distribution Function. Same thing as **DRAW**= .

EXPON/**NOEXPON** specifies that the random number generated is to follow the exponential distribution ($f(x) = \lambda \exp(-\lambda x)$, where the expectation of x is $1/\lambda$, variance = $1/\lambda^2$). Use LAMBDA= to specify λ.

GAMMA/**NOGAMMA** specifies that the random number generated is to follow the gamma distribution ($f(x) = (\lambda^\alpha/\mathrm{gamfn}(\alpha))\, x^{(\alpha-1)} \exp(-\lambda x)$, where the expectation of x is α/λ, and the variance = α/λ^2). Do not use LAMBDA= to specify λ; use the MEAN= and STDEV= options instead. The MEAN and STDDEV must be non-negative.

GENERATOR = 1 or [2]. Type of uniform random number generator. Use GEN=1 to reproduce results from TSP 4.2 to 4.4; this generator has period equal to 2**31 **-** 1. The new (default) generator has period 2**319.

LAMBDA = the exponential or double exponential parameter. (See the EXPON and LAPLACE options).

LAPLACE/**NOLAPLACE** specifies that the random number generated is to follow the Laplace (double exponential) distribution ($f(x) = \lambda \exp(-2|\lambda x|)$, variance = $1/(2\lambda^2)$).

MEAN = the expected value of the random variable. Applies only to the normal, gamma, Poisson, and negative binomial random variables. The default value is one (which you will probably wish to override for the gamma and negative binomial distributions).

NEGBIN/**NONEGBIN** specifies that the random number generated is to follow the negative binomial(N,p) distribution, which is excess waiting time to obtain N successes (the number of trials minus N) with success probability p for each trial. N does not have to be an integer. The mean = $N*(1-p)/p$ and the variance = $N*(1-p)/p^2$ = Mean/p. For this parametrization, the mean can also be written λ/δ , and the variance $\lambda*(1+\delta)/\delta^2$ = Mean*$(1+\delta)/\delta$. Do not use LAMBDA= to specify λ; use the MEAN= and STDEV= options instead. The MEAN must be non-negative, and STDDEV must be larger than or equal to the square root of the mean. See also the **NEGBIN** command.

POISSON/**NOPOISSON** specifies that the random number generated is to follow the Poisson distribution ($f(x) = \exp(-a)\, a^x / x!$). This distribution has one parameter, a, which is both the expected value and the variance. Supply a as MEAN=*value* (a non-negative number).

SEEDIN = value of random seed to start random generator (replaces the current value of the random seed). This must be an integer in the range [1,2.1 billion] (otherwise it is moved into this range). Note that scalar values are stored in double precision, which allows for 16 significant digits, so don't make the seed too large if you are trying to reproduce results. When used with the new default uniform generator, all 8 seeds are set to the SEEDIN value.

SEEDOUT = random seed of the random generator (the current random seed, **before** any random variables are created by this command). To print the seed, use OPTIONS NWIDTH=20; to provide enough digits. SEEDOUT is not useful with the new uniform generator, because it uses 8 seeds.

STDEV = the standard deviation of the random variable. This option applies only to normal, gamma, and negative binomial random variables. The default value is one.

T/**NOT** specifies that the random number generated is to follow student's t distribution with DF=n degrees of freedom.

UNIFORM/**NOUNIFORM** specifies that the random number generated is to follow the uniform distribution between zero and one.

VCOV = (symmetric) variance-covariance matrix for multivariate normal random variables. You can also supply the (triangular) square root of this matrix, which saves a step.

VMEAN = mean vector for multivariate normal random variables (the default is a vector of zeroes).

Examples:

These examples each generate a thousand random numbers and store them under the series name given.

```
SMPL 1 1000 ;
RANDOM STDNORM ;
RANDOM(UNIFORM) FLAT ;
FLATAB = FLAT*(B-A) + A ;            ?  UNIFORM ON THE INTERVAL (A,B).
CDF(CHISQ,DF=3,INV) FLAT CHI3;   ? Chi-square(3)
T1EV = -LOG(-LOG(FLAT));             ? Type I Extreme Value
RANDOM(CAUCHY) FAT ;
RANDOM(EXPON,LAMBDA=2) Z ;
RANDOM(LAPLACE, LAMBDA=0.5) DE ;
RANDOM(T,DF=5) FATAIL ;
RANDOM(GAMMA,MEAN=10,STDEV=2) GAMMA10 ;
RANDOM(NEGBIN,MEAN=10,STDEV=5) NEGBIN10 ;
RANDOM(MEAN=10,STDEV=3.1623) NORM10 ;
RANDOM(MEAN=10,POISSON) POISS10 ;
```

The last two examples produce random numbers with the same mean and variance, but different distributions. A normal variable with mean 10 and standard deviation 3.1623 could also have been generated by the following:

```
NORM10 = 10 + STDNORM*3.1623 ;
```

The next example generates a bivariate normal random vector with correlation .5:

```
LOAD (TYPE=SYM,NROW=2) COVMAT ; 1 .5 1 ;
RANDOM (VCOV=COVMAT) NU1 NU2 ;
```

Now we use the estimated residuals from a regression to generate 10 samples with the same EDF:

```
LIST RES RES1-RES10 ;
RANDOM (EDF=@RES) RES ;
```

When we are done, the list RES consists of ten series which have the same distribution as the original residual series @RES.

Example with inverse distribution functions:

```
RANDOM(SEED=94298,UNIFORM) U ;
EV = -LOG(-LOG(U)) ;   ? Type I Extreme Value
CDF(INV,CHISQ,DF=n) U CHIV ;   ? Chi-square (n)
```

An example using the matrix version of DRAW to draw data from an empirical distribution function in order to investigate the potential rate of convergence of a particular estimator.

```
? original data set has 100 observations, compute residuals
SMPL 1 100 ;
OLSQ Y C X ;
UNMAKE @COEF A B ;
```

```
MMAKE EDF  @RES  X ;
? estimation using 1000 obs drawn from EDF
SMPL 1 1000 ;
RANDOM (DRAW=EDF) E  X ;
Y = A+B*X+E ;
OLSQ Y C X ;
? estimation using 10,000 obs drawn from same EDF
SMPL 1 10000 ;
RANDOM (DRAW=EDF) E X ;
Y = A+B*X+E ;
OLSQ Y C X ;
```

Verify that the new uniform generator is properly implemented (check sum of first 10,000,000 r.v.s for seed 12345):

```
OPTIONS DOUBLE MEMORY=5;
SMPL 1,100000;
RANDOM(GEN=2,SEEDIN=12345);
SET TOTAL=0;
DO I=1,100;
    RANDOM(UNIFORM) X;
    MAT TOTAL = TOTAL + SUM(X);
ENDDO;
SET CORRECT = 5000494.15;
PRINT TOTAL, CORRECT;
```

Method:

The method used by RANDOM is the multiplicative congruential method. The new uniform generator has period 2^{319}, and is a combination of 2 multiple recursive generators with 8 seeds. See L’Ecuyer(1999), generator MRG32k5a.

The old uniform generator has period 2^{31}-1 and multiplier 41358; this choice is described in L'Ecuyer(1990) (it has optimal "randomness" in its class). It is implemented in integer math for speed, and to insure a full period (no repeats in 2^{31} draws).

The multiplicative congruential method produces random numbers which are uniform on the (0,1) interval. Normal and Poisson random variables are created from uniform r.v.s with ACM Algorithms #488 and #369, respectively. Gamma random variables use ACM Algorithm #599. Negative binomial r.v.s are computed by drawing Gamma(δ, λ) random.variables Y, and then drawing Poisson(Y) random.variables. All other random variables are derived from the uniform random variables using the inverse distribution function, which usually involves an asymptotic expansion (see the CDF references).

References:

Efron, Bradley,   "Bootstrap Methods: Another Look at the Jackknife," **Annals of Statistics** 7 (1979), pp. 1-26.

Efron, Bradley, **The Bootstrap, the Jackknife and Other Resampling Plans**, Philadelphia: SIAM, 1982.

Efron, Bradley, and G. Gong, "A Leisurely Look at the Bootstrap, Jackknife, and Cross-validation," **American Statistican**, February 1983, 37(1), pp. 36-48.

L'Ecuyer, Pierre, "Good Parameter Sets for Combined Multiple Recursive Random Number Generators," **Operations Research** 47, 1999.   http://www.iro.umontreal.ca/~lecuyer/papers.html

L'Ecuyer, Pierre, "Random Numbers for Simulation," **Communications of the ACM**, October 1990, pp. 85-97.

Schaffer, Henry E., Algorithm #369, **Collected Algorithms from ACM Volume II**, ACM, New York, N.Y..