The \textit{uniform()} function generates random draws from a uniform distribution between zero and one ([P functions]). One of its many uses is creating random draws from a discrete distribution where each possible value has a known probability.

A uniform distribution means that each number between zero and one is equally likely to be drawn. So the probability that a random draw from a uniform distribution has a value less than .50 is 50\%, the probability that such a random draw has a value less than .60 is 60\%, etc. The example below shows how this can be used to create a random variable, where the probability of drawing a 1 is 60\% and the probability of drawing a 0 40\%. In the first line random draws from the uniform distribution are stored in the variable \texttt{rand}. Each case has a 60\% probability of getting a value of \texttt{rand} that is less than .60 and a 40\% probability that it receives a value more than .60. The second line uses this fact to create draws from the desired distribution. Using the \texttt{cond()} function (Kantor and Cox 2005) it creates a new variable, \texttt{draw}, which has the value 1 if \texttt{rand} is less than .6 and 0 if \texttt{rand} has a value more than .60.

\begin{verbatim}
gen rand = uniform() 
gen draw = cond(rand < .6, 1, 0)
\end{verbatim}

The same result can be achieved with one line of code by using the fact that in Stata a true statement is represented by 1 and a false statement by 0 (Cox 2005). If Stata is given the the following command, it will for each case draw a random number from the uniform distribution, look if that number is less than .6, and if that is true it will give that case the value 1 (true) on the variable \texttt{draw}, and otherwise it will give that case the value 0 (false) on that variable.

\begin{verbatim}
gen draw = uniform() < .6
\end{verbatim}

The probability does not have to be constant. For instance, in the example below the probability of drawing a 1 depends on the variable \texttt{x}. It simulates data for a logistic regression with a constant of -1 and an effect of \texttt{x} of 1. In this example the variable \texttt{x} consists of draws from a standard normal distribution.

\begin{verbatim}
gen x = invnorm(uniform()) 
gen draw = uniform() < invlogit(-1 + x)
\end{verbatim}
Stata tip 48

Nor is this method limited to random variables with only two values. The example below draws from a distribution where the value 1 has a probability of 30%, the value 2 a probability of 45%, and the level 3 a probability of 25%.

```stata
gen rand = uniform()
gen draw = cond(rand < .3, 1, /*
   */ cond(rand < .75, 2, 3 ))
```

This same principle can be used to create draws from a binomial distribution. Remember that a binomial distribution with parameters \( n \) and \( p \) is the distribution of the number of 'successes' out of \( n \) trials when the probability of success in each trial is \( p \). One way of sampling from this distribution is to literally do just that, i.e. draw \( n \) numbers from a uniform distribution, declare each number a success if it is less than \( p \), and then count the number of successes (Devroye 1986, p. 524). In this case it is convenient to use Mata and the Mata equivalent of \( \text{uniform()} \), \( \text{uniform(r,c)} \), which creates an \( r \) by \( c \) matrix filled with random draws from the uniform distribution. The example below creates a new variable `draw` containing draws from a binomial\((100,.3)\) distribution:

```mata
mata:
n = 100
p = .3
draw = J(st_nobs(),1,.) // matrix to store results
for(i=1; i<=rows(draw); i++) { // loop over observations
   trials = uniform(1,n) // create n trials
   successes = trials :< p // success = 1 failure = 0
   draw[i,1] = rowsum(successes) // count the successes
}
idx = st_addvar("int", "draw")
st_store(.,idx,draw) // store the variable
end
```

References


