From The Edge:

by

A. Razzak Memon

Using RANDOM Function in R:BASE

February 19, 2003

From the Edge:	Using RANDOM Function in R:BASE
Section:	Functions
Chapter:	Running R:BASE Your Way!
Platform:	R:BASE for DOS/Windows
Version:	6.5 and Higher

R:BASE supports random number generation with the function RANDOM and with the internal UDF called _RANDOM.

At the low level they both call the compiler library routine called RAND. This function generates what are called pseudorandom numbers.

Here is documentation taken directly from the MSDN library:

“The srand function sets the starting point for generating a series of pseudorandom integers. To reinitialize the generator, use 1 as the seed argument. Any other value for seed sets the generator to a random starting point. Rand retrieves the pseudorandom numbers that are generated. Calling rand before any call to srand generates the same sequence as calling srand with seed passed as 1.”

Within R:BASE we do not have an function called srand to set the seed, but if you call RANDOM with a negative argument, R:BASE uses the absolute value of that argument internally to call srand. When you call the RANDOM function, its argument, if positive, specifies the upper range limit you want for the random integer numbers generated. If you do not want to always have the same seed, a good choice is to pass a number based on the current time. Here is some R:BASE code to do that:

```
*(set the seed value)
SET VAR vSeed INTEGER = (.#time - 00:00:00)
SET VAR vSrand = (RANDOM(-1*.vSeed))

*(get some random numbers)
SET VAR v1 INTEGER = (RANDOM(100))
SET VAR v2 INTEGER = (RANDOM(100))
SET VAR v3 INTEGER = (RANDOM(100))
```

The internal UDF functions that support random number use the same library routine, but the final result is a text value since all UDFs return text strings. To set the seed in the UDF version, we actually do have another internal UDF for that. Here is the equivalent for the above code using the UDF functions. The internal UDF _RANDOM function returns numbers up to the maximum for integer values, regardless of the parameter passed to the UDF)

```
*(set the seed value)
SET VAR xSeed INTEGER = (.#time - 00:00:00)
SET VAR xSrand = (UDF('_SRAND', (-1*.xSeed)))

*(get some random numbers)

SET VAR x1 = (UDF('_RANDOM', '0'))
SET VAR x2 = (UDF('_RANDOM', '0'))
SET VAR x3 = (UDF('_RANDOM', '0'))
```