SIMULATION PROGRAMS FOR THE TI-84+ OPERATING SYSTEM
COMMON CORE ALGEBRA II

The PARCC EOY Common Core Standards for Algebra II in the Statistics domain include references to using **simulation**, specifically to develop the likelihood that an observation could exist by random chance alone. The specific standards are shown below.

STATISTICAL STANDARDS CALLING FOR SIMULATION

S-IC.2 Decide if a specified model is consistent with results from a given data-generating process, e.g., **using simulation**. *For example, a model says a spinning coin falls heads up with probability 0.5. Would a result of 5 tails in a row cause you to question the model?*

S-IC.4 Use data from a sample survey to estimate a population mean or proportion; develop a margin of error through the **use of simulation models** for random sampling.

S-IC.5 Use data from a randomized experiment to compare two treatments; **use simulations** to decide if differences between parameters are significant

But, how do we do this simulation with students in a technologically appropriate manner? We've developed three simulation programs for the Texas Instruments 84+ (TI-84+) operating system to do the major types of simulation that we believe is necessary. More may be added in the future. The description of each program is given below and the code is on the pages to follow. We hope to have a more convenient way for you to access this code, but it can be manually entered into your calculator and then easily transferred to others through LINKS.

RANDSAMP: This program allows the user to take a simple random sample from a population. The user inputs the population values into L_1. The user then has two options, a single sample from the population or multiple simulations. In one case, you get a random sample of whatever size you choose drawn from the population (without replacement). In the multiple simulations, the user specifies the sample size and number of samples (simulations). The calculator then stores the sample means in a separate list. This is particularly helpful in testing whether a particular sample mean could have come from a population with a given mean and standard deviation.

PSIMUL: This program is very simple. It takes a proportion for a population and then generates a sample from this population of user specified size. It does this with a simple random sample **with replacement**. So, the population size is large compared to the sample size. The user specifies how many samples from this population should be taken. The program then calculates the sample proportions and stores them in a separate list for all of the simulations. You can use this program to see if a sample of a with a particular proportion could have come from a population with a different proportion. You could even address the fair coin question by giving the population a proportion of 0.5 and the sample size would be the number of flips you make of the coin.

MEANCOMP: This program allows the user to test if two treatments resulted in differences of means that can be attributed to the treatments. The user puts the two sets of results in L_1 and L_2. For example, say you grew 20 plants in the sunshine and 30 plants in the dark. The final heights would be put in these two lists. The program then throws these two lists together, randomly scrambles them up, and redistributes them into two new lists (of the same size as the original two). The differences in the means of the new lists are calculated. The user can specify the number of simulations. By doing this a user can establish how likely the difference in the original two means would be by random chance alone, thus establishing the confidence (or lack thereof) that the original difference is statistically significant.

RANDSAMP

```
:ClrHome
:ClrList L2, L3, L4, L5
:dim(L1)→P
:Disp "1.SINGLE SAMPLE"
:Disp "2.MULTIPLE SAMPLE"
:Input A
:If A=1
:Then
:Disp "SAMPLE SIZE"
:Input N
:seq(X, X, 1, P) → L2
:seq(rand, X, 1, P) → L3
:SortA(L3, L2)
:For (I, 1, N)
:L1(L2(I)) → L4(I)
:End
:Disp "SAMPLE IN L4"
:Else
:ClrHome
:Disp "SAMPLE SIZE"
:Input N
:Disp "NUMBER OF SIMULATIONS"
:Input C
:For (I, 1, C)
:ClrHome
:Disp "SIMULATION", I
:seq(X, X, 1, P) → L2
:seq(rand, X, 1, P) → L3
:SortA(L3, L2)
:For (J, 1, N)
:L1(L2(J)) → L4(J)
:End
:mean(L4) → L5(I)
:End
:SortA(L5)
:Disp "SAMPLE MEANS IN L5"
:End
```

For run **RANDSAMP** simply put your population data in L_1. The program will then direct you to choose a single random sample of whatever size from the population or multiple samples from the population. It will calculate sample means and store them in a separate list.

PSIMUL

```
:ClrHome
:ClrList L1
:Disp "POPULATION P"
:Input P
:Disp ""
:Disp "SAMPLE SIZE"
:Input N
:Disp ""
:Disp "SIMULATIONS"
:Input T
:ClrHome
:For (B, 1, T)
:Disp "SIMULATION", B
:0 → S
:For (I, 1, N)
:If rand ≤ P
:Then
:S+1 → S
:End
:End
:Disp ""
:Disp "SAMPLE P",round(S/N, 2)
:round(S/N, 2) → L1(B)
:End
:SortA(L1)
```

PSIMUL allows you to take a sample of any size from a population of known proportion. This sample will be done with replacement. The assumption is that the population is very large in size compared to the sample. Sample proportions will be displayed as the simulations progress and stored in a separate list.

MEANCOMP

```
:ClrHome
:ClrList L3, L4, L5, L6
:dim(L1) → M
:dim(L2) → N
:1 → Z
:Disp "SIMULATIONS"
:Input S
:ClrHome
:Disp "SIMULATION=",1
:For (Q, 1, S)
:seq(X, X, 1, M+N) → L5
:seq(rand, X, 1, M+N) → LRAND
:SortA(LRAND, L5)
:For (A, 1, M)
:If L5(A) > M
:Then
:L2(L5(A)-M) → L3(A)
:Else
:L1(L5(A)) → L3(A)
:End
:End
:For (A, M+1, M+N)
:If L5(A) > M
:Then
:L2(L5(A)-M) → L4(A-M)
:Else
:L1(L5(A)) → L4(A-M)
:End
:End
:ClrHome
:Disp "SIMULATION=", Z+1
:mean(L4)-mean(L3) → L6(Z)
:Z+1 → Z
:End
:ClrHome
:Disp "DIFFS IN L6"
```

To run **MEANCOMP** place the data from your two treatments in L_1 and L_2. Note that the treatments do **not** need to have the same number of results. The program will randomly scramble these two sets of results and redistribute them to two new lists. Differences in treatment means will then be calculated and stored in a separate list.

Note: The L in front of the variable RAND establishes a list name. So I know have a list named RAND that I can work with. Not to be confused with the rand operator.

You find the L by going to LIST then OPS. It is towards the bottom of this submenu.