

Prof. David Draper
Department of
Applied Mathematics and Statistics
University of California, Santa Cruz

AMS 7L: Lab 3

The lab this week is devoted to a demonstration of the *Central Limit Theorem* (CLT), which says informally that if you take an IID sample of size n from a population of size N and calculate the sum or mean of the sampled draws, as long as n is large the distribution (in repeated sampling) of the sum or mean will be approximately normal¹. This is an extremely useful theorem for a number of purposes in probability and statistics, but (like all limit theorems) it's regrettably silent on the important implementational question of how big n has to be to get a good normal approximation. Theory says that this depends on the population distribution — the closer the population histogram is to the normal curve to begin with, the smaller n has to be to get a good normal approximation for the sum or mean of n IID draws from that population — but this still may leave you wondering, for the population you're working with, about whether the CLT applies with the sample size n you're planning to use.

I've written two programs (in JMP these are called *scripts*, and they're written in what the JMP people call *JSL* [JMP script language]), one for sums and one for means, to simulate the process of repeatedly drawing n times via SRS from a population and collecting the sums or means together, so that we can experimentally explore this issue.

JMP knows how to compute two numerical descriptive summaries that are useful in studying the CLT: *skewness* and *kurtosis*. Skewness is a numerical measure of the extent to which a histogram has a long left- or right-hand tail; histograms with a long left-hand tail have negative skewness, symmetric histograms have skewness 0, and histograms with a long right-hand tail have positive skewness. Kurtosis is a numerical measure of how heavy the tails are for a symmetric histogram, and you have to check to see how it's defined in the book or computer program you're using: with the definition used by JMP, the kurtosis of any normal distribution is 0, histograms with lighter tails than the normal have negative kurtosis, and histograms with heavier tails than the normal have positive kurtosis. The reason skewness and kurtosis are useful in this lab is that we're going to be judging when the sample size n is big enough for the Central Limit Theorem (CLT) to give us a good normal approximation to the long-run histogram of the sum S or mean \bar{y} in the imaginary data set, and — in addition to the graphical methods we've already talked about (histograms, outlier box plots, stem-and-leaf plots) — it's helpful to have informal numerical methods as well (from the descriptions above, you can see that you want n to be big enough so that the skewness and kurtosis of the imaginary data set are both close to 0).

Another helpful thing to remind you of: you can use JMP to superimpose on the histogram of the imaginary data set both the best-fitting normal curve and the best-fitting density curve (as noted in Lab 1, you get this in JMP with the **Smooth Curve** option of the **Fit**

¹And we already know from common sense that if N is a lot bigger than n , IID sampling should be about the same as SRS, so the theorem applies to SRS in that case too.

Distribution menu that's part of the output of the Distribution option of the Analyze menu) — the idea is that when a data set is close to normal the best-fitting normal curve and the best-fitting density curve should be almost the same.

1. **Using JMP scripts.** Go to the AMS 7/7L course web page:

`www.soe.ucsc.edu/classes/ams007/Fall106`

Toward the bottom, in the section called Data files (in .jmp format) and scripts (in .jsl format) for the labs:, there's a new entry that refers to the Central Limit Theorem. Download copies of the two scripts

```
ams7L-random-sums.jsl
ams7L-random-means.jsl
```

and all 6 of the datasets

```
ams7L-roulette1.txt
ams7L-uniform.txt
ams7L-fake.txt
ams7L-sedge.txt
ams7L-roulette2.txt
ams7L-roulette3.txt
```

to the desktop (each of these is a text file; to save them, pull down the File menu on your browser and select Save Page As (Ctrl S is a short-cut for these operations), make the Save As window point to your desktop, and click Save). Use the instructions from Lab 1 or 2 (double-click on PC Server on 'be-lab' (F), double-click on Math, Statistics and Graphing, and double-click on JMP6 shortcut) to enter JMP. In the JMP Starter window click Open Script, make the Open Data File window point to your desktop, and double-click on ams7L-random-sums; then foreground the JMP Starter window, click on Open Data Table, make the Open Data File window point to your desktop if it's not already pointing there, under Files of type: select Text Import Files (this is how to bring data sets in .txt format into JMP), and double-click on ams7L-roulette1. Minimize the JMP Starter window, foreground the ams7L-random-sums window, and resize this window by making it wide enough so that you can look at the script; read the comment at the beginning and move down far enough in this file so that you see the lines

```
n=50; // size of sample drawn from population;
```

and

```
M=500; // number of simulated sums in imaginary data set;
```

In what follows you get to play with three ingredients:

- the population data set (which is always the current data table, and the column called **Y** in this data table is the population; the current data table above is `ams7L-roulette1`); in all cases I made the population size N huge (around 100,000) so that SRS is approximately the same as IID;
- the sample size n ; and
- the number M of simulated values in the imaginary data set.

The theory behind the CLT assumes that M is enormous (in fact, the formulas for the expected value and standard error correspond to $M = \infty$). If we had enough time or contemporary computers were a lot faster we might work with an M of 10,000 or more; I've set $M = 500$ in the scripts, which is big enough to get a decent initial idea of what's going on (I've encouraged you to make M bigger when you do the assignment); the idea will be for you to look at various population data sets and figure out how big n needs to be with each of these populations to get the CLT to work for you.

As a first example of using the scripts, with `ams7L-roulette1` as your only data set in the JMP window, foreground the `ams7L-random-sums` script, make sure that `n=50` and `M=500` are specified in the script, and from the **Edit** pull-down menu click on **Run Script** (**Ctrl R** is the short-cut for this) — JMP will create a new data table called **Results** and will take a few minutes to fill it with $M = 500$ randomly-simulated sums of $n = 50$ approximately IID draws from the population that corresponds to betting \$1 on a single number at roulette. Use the **Analyze** capabilities on the **Sum** column in the **Results** data table to see if $n = 50$ is big enough to get a good normal approximation to the distribution of the sum of draws from this population; in particular the histogram, outlier box plot, and normal quantile plot should help you to see that $n = 50$ does not produce a very good normal approximation, and the skewness and kurtosis are nowhere near 0. To try again with a bigger value of n , close the **Results** window (JMP will ask you if you want to save changes in it, and you can say no), foreground the `ams7L-random-sums` script, change the line that says `n=50` to read (for example) `n=100`, and repeat the steps above. You should find that as you increase n the normal approximation to the sum will improve — this is the CLT in action.

There will be a handout in class on the CLT in which the populations `uniform` and `fake` are examined — try the script `ams7L-random-means` on `uniform` and `ams7L-random-sums` on `fake` with various values of n ; you should find that much smaller values of n give good normal approximations with this population than with `roulette1` (**Note:** to run the scripts, only one data table containing a column called **Y** can be displayed in the JMP window; otherwise the script will get confused about which **Y** you want it to sample from). The population `sedge` is based on the number of plants per quadrat found in a study by Archibald (1950) of a type of sedge (these are plants that looks a lot like grasses or rushes) called *Carex flacca*; this population has a very long right-hand tail. The populations `roulette2` and `roulette3` are set up to model two other types of wagers you can make at roulette: betting on a double number (`roulette2`) or betting on red versus black (`roulette3`).

Assignment for Lab 3. For each of the 6 data sets (`roulette1` (single-number), `uniform`, `fake`, `sedge`, `roulette2` (split), and `roulette3` (red versus black)), thinking in turn of each

of them as a population to be randomly sampled from, set M to at least 1,000 (for greater accuracy of your simulations) and figure out what is the smallest value of n for each of these populations to get a good normal approximation for the (repeated-sampling) distribution of the sum or mean (it doesn't matter which script you use); use the skewness and kurtosis values and histograms with the normal curve and the smooth density curve superimposed on them. For each population your write-up should show JMP numerical and graphical descriptive results for some values of n that are too small and for the smallest n that you think gives a good normal approximation. **Extra credit:** Run the scripts on the 3 roulette populations and use the results to estimate the probability of coming out ahead with $n = 1,000$ \$1 bets on each of (a) a single number, (b) a split and (c) red versus black.

This assignment is due in lab next week.