University of Essex
2015-16

Department of Government
TA: Stavros Poupakis

GV903: Advanced Research Methods

Lab 3: Random Variables

Suppose you want to generate random numbers from a distribution of your choice, but you only have a random number generator for the Uniform distribution $u \sim U(0,1)$. If you want to draw from a Uniform distribution all you need to do is

```
clear all
set obs 1000
gen u = runiform(0,1)
hist u, bin(10) frequency
```

Draw from a Uniform distribution for different samples of size 10, 100, 1000 and 1000000 and plot the histograms (use 10 bins). What do you observe?

What if we want to draw from another distribution. Still this is possible, thanks to inverse probability sampling (inverse probability integral transform).

Suppose we want to draw from an exponential distribution $X \sim \text{Exp}(\lambda)$, with pdf $\lambda e^{-\lambda x}$. Then,

$$f(x) = \lambda e^{-\lambda x}$$
$$F(x) = 1 - e^{-\lambda x}$$
$$e^{-\lambda x} = 1 - F(x)$$
$$-\lambda x = \ln(1 - F(x))$$
$$x = -\frac{\ln(1 - F(x))}{\lambda}$$

Letting $F(x) = u$ with $u \sim U(0,1)$, we can generate x to be $X \sim \text{Exp}(\lambda)$, as

$$x = -\frac{\ln(1-u)}{\lambda} \quad \text{or even} \quad x = -\frac{\ln(u)}{\lambda}$$

Getting back to Stata, now that we know the math, this is

```
set obs 1000
gen e = -ln(runiform(0,1))/1
```

Try to plot the histogram of this and also the Kernel density plot. Makes sense?

So good news, we can generate any continuous distribution we want from a uniform. We can do the same for discrete with a similar, even easier, approach. However, Stata provides us with many ready built-in commands to generate other distributions, so we don't need to do it on our own. Here are some examples:

```
rnormal(0,1)
rpoisson(1)
rt(50)
rchi2(1)
```

Generate variables from these distributions ($n = 10,000$) and see their histograms.
In the lecture you discussed the Lindberg-Levy CLT, let's refresh the definition:
If $x_1, x_2, x_3, ..., x_n$ are a random sample from a probability distribution with finite mean μ and finite variance σ^2 and $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$, then

$$\sqrt{n}(\bar{x}_n - \mu) \xrightarrow{d} N(0, \sigma^2)$$

To see that in Stata. First, we need to write a *program* which will enable us to perform a task fast again and again - and this is the basic reason of writing programs! Type the following in your do-file:

```
program define my_first_program, rclass
        drop _all
        set obs 50
        gen x = rpoisson(1)
        sum x
        return scalar x_mean = r(mean)
end
```

Then simply call the *program* by typing its name `my_first_program` and run it in Stata.
Try to do that a few times, what do you observe?

Now let's demonstrate our simulation. To do that, we will repeat what you just did for m times, where m should be as big as 2,000 (in fact in whenever you do Monte Carlo simulations you should use 10,000). Run the following:

```
simulate xbar=r(x_mean) , reps(2000): my_first_program
```

Find out the mean and the variance of the new variable that has been created and plot its histogram. What do you observe? What if we repeat this exercise taking a larger sample, say 1,000 or 10,000?

Now generate another version of this variable by typing the following:

```
sum xbar
gen z =  sqrt(50)*(xbar-r(mean))
```

Does this remind you anything? Find out the mean and variance and plot the histogram.

What you just did is what people call Monte Carlo simulations. MC simulations are used for two main reasons, one to understand what is actually happening with the assumptions make in our methods and second to investigate the finite sample properties of various statistics.
Try to repeat this exercise for the Uniform distribution $U(0,1)$ and the Chi-squared with 1 degrees of freedom χ^2.