- HW #8 is due on Friday, March 30 at 2:30pm.
- All homework must be turned in electronically from now on.
- There is a lot of computing on this assignment; please let me know early if there will be coding challenges!

Notes: We discussed the fact that this course will essentially cover Chapters 1–6 and 11 in Ross, plus a few additional topics in Monte Carlo methods and MCMC. Nothing that is not covered in the course will be on the statistics qualifying review exam.

$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} g(X_i)$, where X_i are i.i.d. from f.

- Why this works: (Strong or weak) law of large numbers
- More precise information is provided by the Central Limit Theorem.
- The above results are true for vectors X_i of arbitrary dimension.
- *Thinking about i.i.d. Monte Carlo $\equiv$ thinking about basic statistics.*

Notes: We talked a bit more about the CLT, in particular the exact statement of the theorem, what it means, and how we can use it to give an approximate distribution of the $\hat{\mu}$ random variable:

$$\hat{\mu} \textit{ is approximately } N\left(\mu, \frac{\mathrm{Var}_f\, g(X)}{n}\right)$$

Toy example from Charlie Geyer via Murali Haran:

- Given $X_1, X_2 \overset{\text{iid}}{\sim} N(0,1)$, what is $P(X_2 < X_1^2)$?
- Try numerical integration.
- Try Monte Carlo approach.
- Relative advantages / disadvantages?

Notes: We went through this example in R, using both the numerical integration and Monte Carlo approaches. In this case, numerical integration appears preferable, but this may not always be the case, for a couple reasons. The R code is available online (see the calendar for today's class).

How do we get "random" numbers from an arbitrary distribution F?

- Everything starts with uniform pseudo-random numbers.
- "Uniform random" numbers are neither (!!)
- Refer to `help(Random)` in R.
- Generating pseudo-random (and pseudo-i.i.d.) numbers is a huge topic that we will not explore in depth.

Notes: We looked at the `help(Random)` *page in R, discussing the fact that pseudo-random numbers are not at all random but are instead completely deterministic. However, many very smart people have developed numerous routines over the last several decades to make the generated numbers appear as "random" and "independent" as possible.*

Properties of uniform random number generators:

- Repetititve
- Not independent
- Not even continuous

These are not always disadvantages!

Notes: The period of a random number generator is the number of values that must be simulated before the entire sequence of values repeats itself. For any good modern algorithm, the period is much larger than the number of picoseconds in the entire history of the universe! The non-independence can be an asset because you can set the same seed value before generating your random numbers in order to obtain identical results in a repeat of the experiment. Finally, the non-continuity results from the fact that computer arithmetic actually divides the unit interval interval into a very large number of tiny increments (about 10^{-15} *wide), and only integral multiples of the increment are possible.*

How do you generate $X \sim F$ given $U \sim \text{Unif}(0,1)$?

- Some special cases are covered in Ross, section 11.3
- General "inversion" method based on quantile function

$$F^{-}(u) \stackrel{\text{def}}{=} \inf\{x : u \leq F(x)\}.$$

Can prove: $F^{-}(U) \sim F$, i.e., $P[F^{-}(U) \leq x] = F(x)$ for all x.

Notes: A couple important cases to be aware of are the Box-Muller method and the Marsaglia polar method for generating normal random numbers from uniform random numbers. We began discussing the inversion method today and will continue on Wednesday. Of particular importance for understanding the intuition of the inversion method is the fact that when $F(x)$ *is invertible (because it is strictly increasing and continuous),* $F^{-1}(u)$ *is the same as* $F^{-}(u)$.