Random Numbers

- Built-in random integer generator `rand()`
- Do NOT use `rand()`
- Use `drand48()`, which is in the math library
- It is a good linear conguential random number generator, generating random U[0,1]
- Can set the seed with `srand48()`
- Of course, these are really pseudo-random numbers

Other Random Numbers

- U[a,b] from `(b-a)*drand48()+a`
- Uniform on integers from 1 to `k` with `ceil(drand48()*k)`
- Inverse CDF Transformation: To sample X from a distribution with CDF F, sample $U \sim U[0,1]$ and let $X = F^{-1}(U)$.
- Box–Muller transformation for normals
- other transformations (e.g., `-log(drand48())/`$\lambda \sim$ Exp(λ))
- Library Code (e.g., Numerical Recipes)

Box–Muller Transformation

- Generates two independent N(0,1) from two independent U[0,1]
- Idea: let $X_1, X_2 \overset{iid}{\sim} N(0,1)$
- What is the distribution of $X_1^2 + X_2^2$?
- What is the distribution of $tan^{-1}\left(\frac{X_1}{X_2}\right)$?
- (It may help to think of the plot of X_1 vs. X_2.)

- $X_1^2 + X_2^2 \sim \chi_2^2 = Exp(\frac{1}{2})$
- $tan^{-1}\left(\frac{X_1}{X_2}\right) \sim U[0, 2\pi]$
- Thus let $U_1, U_2 \overset{iid}{\sim} U[0, 1]$
- Let $R = \sqrt{-2 * \log(U_1)},\ T = 2\pi * U_2$
- Let $X_1 = R * cos(T),\ X_2 = R * sin(T)$
- You can verify that this works by doing the transformation on the densities and computing the Jacobian.

`randnorm1.c, randnorm2.c`