Opinionated Lessons in Statistics

by Bill Press

#11 Random Deviates

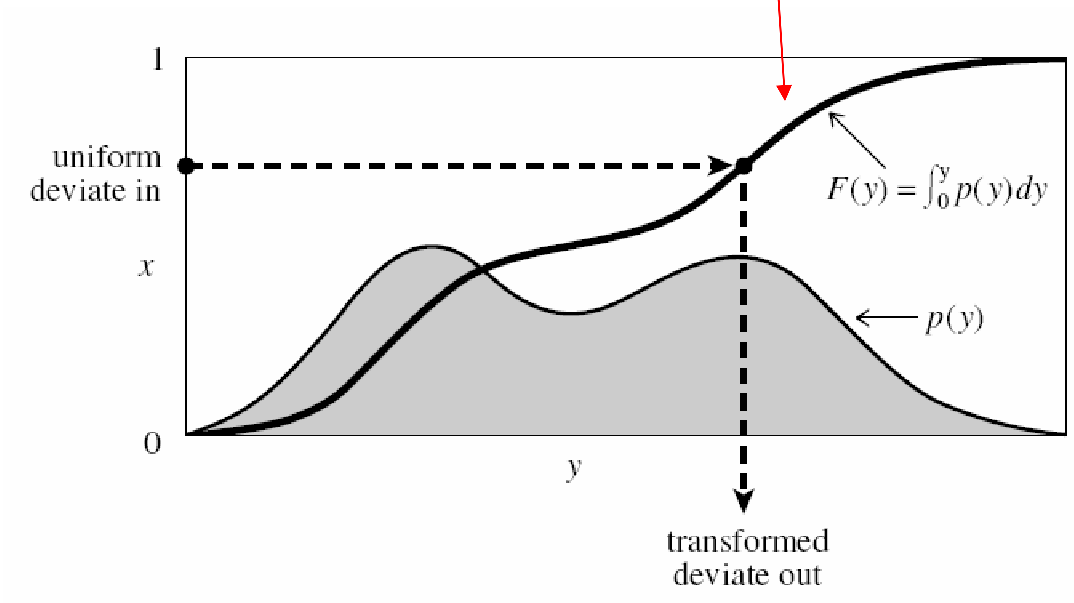
Random deviates from univariate distributions

You generally have a random generator for U(0,1). What do you do for other distributions? Note that generators need to be *fast*, because you often call them millions or billions of times!

Here are three general methods:

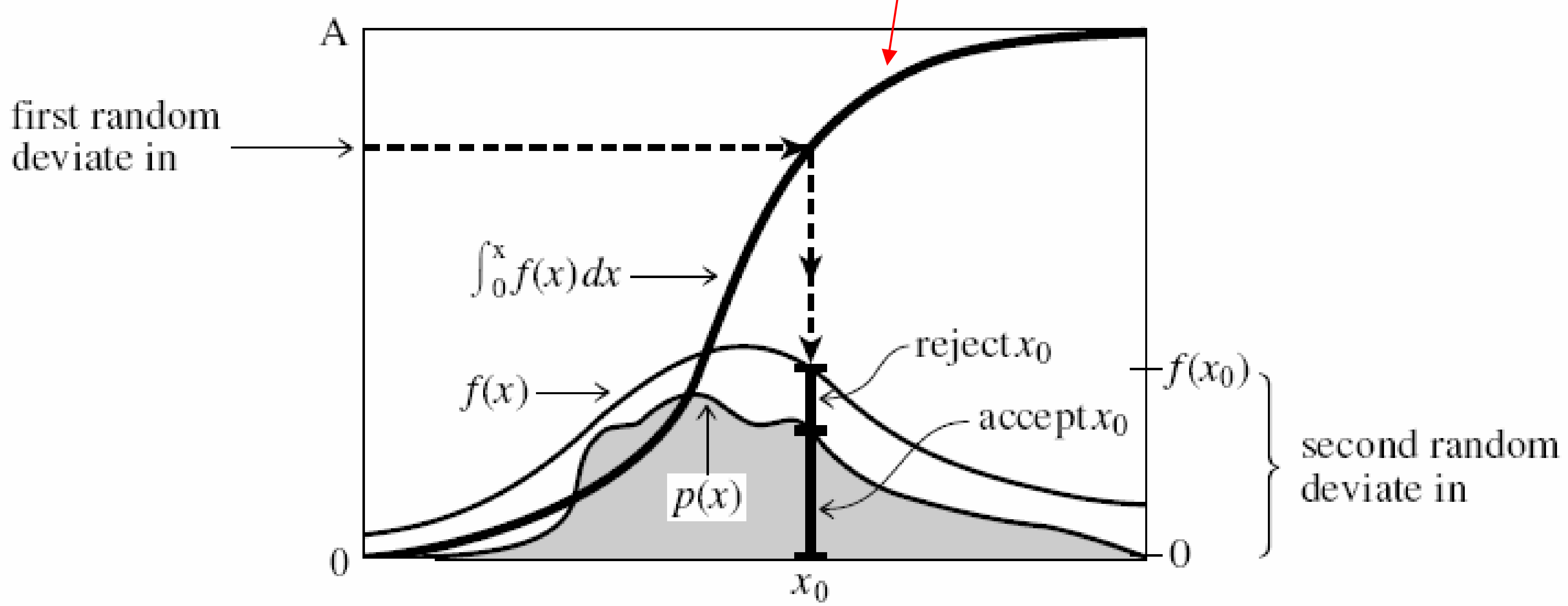
Transformation method:

You must be able to calculate the inverse of the cdf of your distribution!



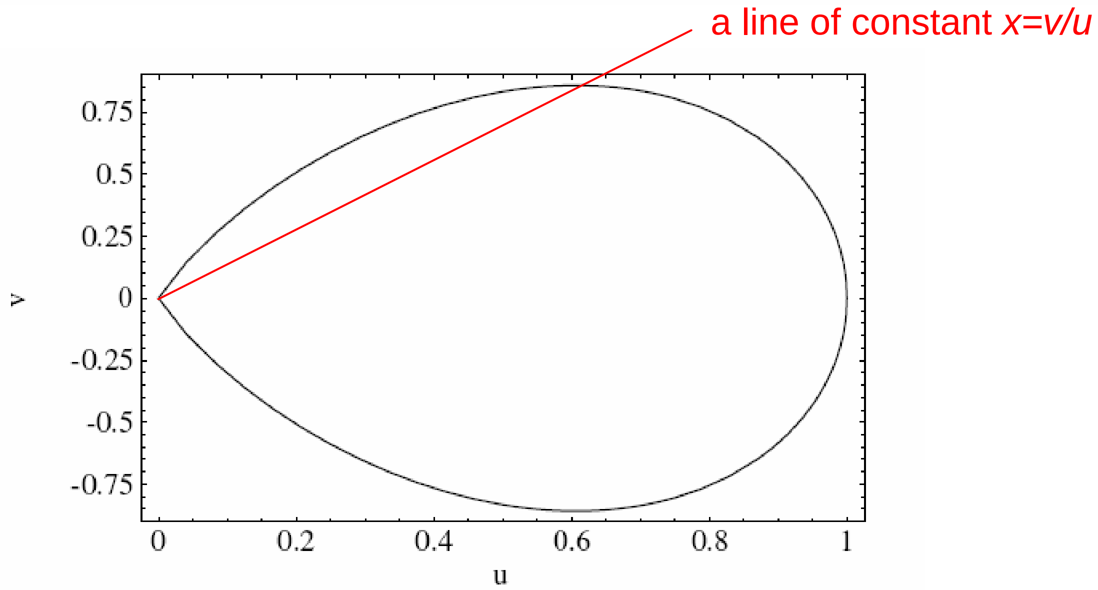
Rejection method:

You must have a bounding function for which you are able to calculate the inverse of the cdf!



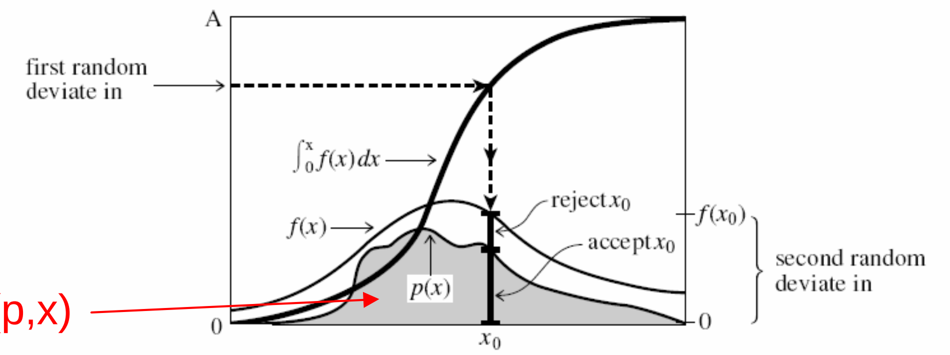
Ratio of Uniforms Method
(some of the best features of both xformation and rejection)

- Construct the region in the (u, v) plane bounded by $0 \leq u \leq [p(v/u)]^{1/2}$.
- Choose two deviates, u and v, that lie uniformly in this region.
- Return v/u as the deviate.

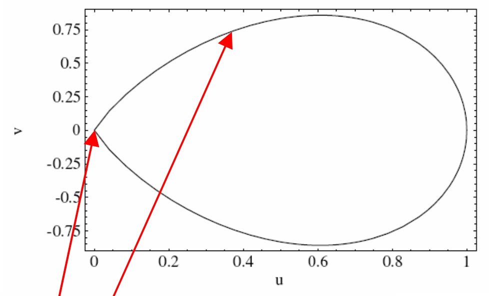


Proof of Ratio-of-Uniforms Method

$$p(x)dx = \int_{p'=0}^{p'=p(x)} dp'\,dx$$



i.e., sample uniformly in the (p,x) plane, in the shaded region



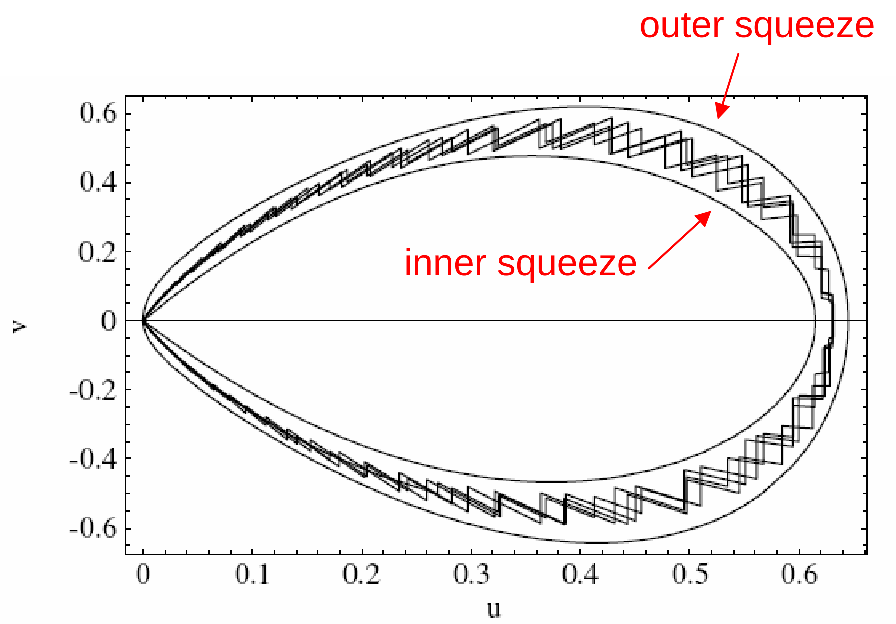
$$\frac{v}{u} = x$$
$$u^2 = p$$

change of variables

$$p(x)dx = \int_{p'=0}^{p'=p(x)} dp'\,dx = \int_{u=0}^{u=\sqrt{p(x)}} \frac{\partial(p,x)}{\partial(u,v)} du\,dv = 2\int_{u=0}^{u=\sqrt{p(v/u)}} du\,dv$$

(be sure you understand Jacobian determinates!)

Ratio of Uniforms is particularly powerful when combined with squeezes



For this particular example the desired distribution is integer valued (binomial deviates), hence the staircases. For a continuous distribution, there would just be a smooth curve between the squeezes (which would typically be too close together to see clearly).

you only compute p(v/u) when you are between the squeezes!

e.g., Leva's algorithm for normal deviates:

```
struct Normaldev : Ran {
Structure for normal deviates.
    Doub mu,sig;
    Normaldev(Doub mmu, Doub ssig, Ullong i)
    : Ran(i), mu(mmu), sig(ssig){}
    Constructor arguments are μ, σ, and a random sequence seed.
    Doub dev() {
    Return a normal deviate.
        Doub u,v,x,y,q;
        do {
            u = doub();
            v = 1.7156*(doub()-0.5);
            x = u - 0.449871;
            y = abs(v) + 0.386595;
            q = SQR(x) + y*(0.19600*y-0.25472*x);
        } while (q > 0.27597
            && (q > 0.27846 || SQR(v) > -4.*log(u)*SQR(u)));
        return mu + sig*v/u;
    }
};
```

here, only ~1% of the *(u,v)* area is between the squeezes, requiring the calculation of the log

That's all for random deviates, a great subject but not this course!