Download codes for this lecture at:

Lecture 4 codes: https://www.dropbox.com/s/3ooonh87b52w9tn/Lec4.R?dl=0

Monte Carlo Integration

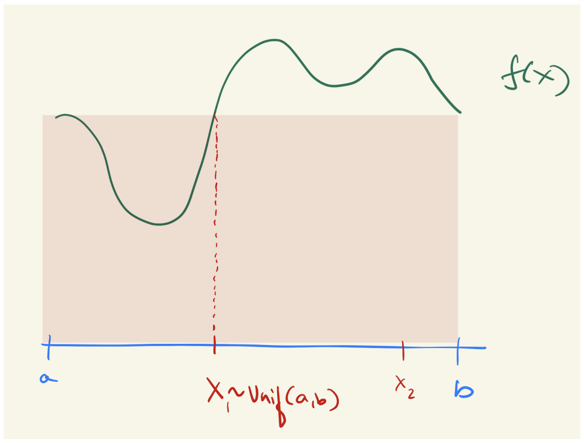
In Riemann Sums, we are essentially fitting a grid on the line, fixing the points on the grid and then making rectangles.

But, what if instead of fixing the points on the grid, we randomly draw uniform points on the line! Consider calculating the integral

$$\theta = \int_a^b f(x)dx\,.$$

Let $X_1 \sim \text{Uniform}(a, b)$. If I give you no other information about the function $f(x)$, and just give you $f(X_1)$, what is your best guess for θ?

It is probably $(b-a)f(X_1)$!



Now, I give you one more draw $X_2 \sim \text{Uniform}(a, b)$. Now, what is your best guess for θ:

1. $(b-a)f(X_1)$?
2. $(b-a)f(X_2)$?
3. $\dfrac{(b-a)f(X_1) + (b-a)f(X_2)}{2}$?

Since you now have two draws and two points of information about the height of the box, you want to use both pieces of information; thus option (3) seems better.

Continuing with this stream of logic, let $X_1, X_2, \dots, X_n \sim \text{Uniform}(a, b)$. Then for each $i = 1, \dots, n$,

$$\hat{\theta} = (b-a)\frac{1}{n}\sum_{i=1}^{n} f(X_i)$$

This process of *estimating* integrals using random draws is called Monte Carlo integration. Let's try and do this for

$$\theta = \int_0^1 \exp(-x^2/20)$$

Convergence

The *law of large numbers* gives a stabilizing behavior of any mean, so that, as $n \to \infty$,

$$\hat{\theta}_n \to \theta\,,$$

where $\hat{\theta}_n$ is the estimate from n samples. So understand this a little mathematically, note that

$$\begin{aligned}
\hat{\theta}_n &= \frac{b-a}{n}\sum_{i=1}^{n} f(X_i) \\
&= \frac{b-a}{n}\left[\sum_{i=1}^{n-1} f(X_i) + f(X_n)\right] \\
&= \frac{n-1}{n}\frac{b-a}{n-1}\left[\sum_{i=1}^{n-1} f(X_i)\right] + \frac{b-a}{n} f(X_n) \\
&= \frac{n-1}{n}\hat{\theta}_{n-1} + \frac{b-a}{n} f(X_n)\,,
\end{aligned}$$

for large n, $(n-1)/n \approx 1$ and the second term is small. Thus, with every new data point added, the change in the estimate tends to be smaller and smaller.

We will visualize this in `R`!