Download codes for this lecture at:

Lecture 3 codes: https://www.dropbox.com/s/z25jx4vyyr1npq2/Lec3.R?dl=0

Riemann Sums

Consider calculating the following integral

$$\theta = \int_a^b e^{-x^2/20} dx \,.$$

This is difficult enough that I do not want to do this via integration tricks. So instead I can use crude Riemann approximations, that we've learned in the Calculus class already.

To approximate the area under the curve, we break the integral ends (a, b) into equal length bins. Let the number of bins be n. Let the lower ends of the bins be x_i, that is for $i = 0, \ldots, n$

$$x_i = \frac{b-a}{n} i + a \,.$$

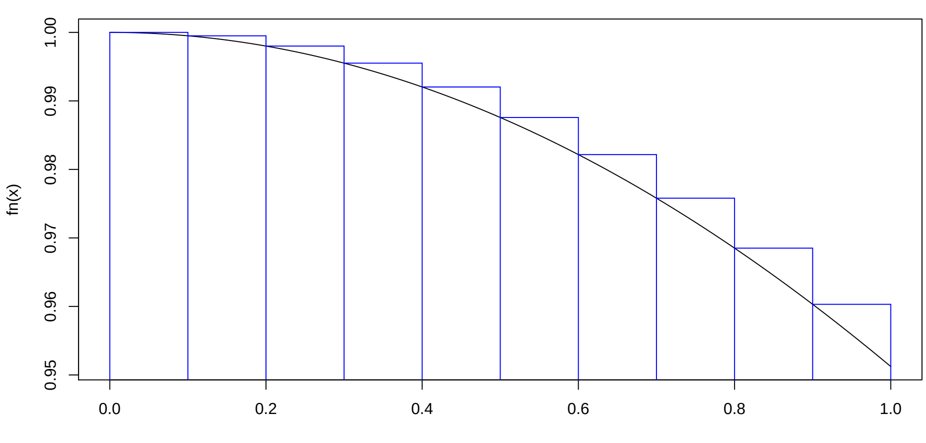
The length of each bin is then

$$h := \frac{b-a}{n} \,.$$

Note that $x_0 = a$ and $x_n = b$. Then,

$$\theta \approx \sum_{i=1}^{n} f(x_{i-1}) \left(\frac{b-a}{n} \right)$$

The figure below is with $n = 10$.



In the calculus class, we've seen that as $n \to \infty$, the Riemann sum converges closer and closer to the truth. We will now show this in R!

```
# Let's make a function for f(x) = exp(-x^2/20)
fn <- function(x)
{
  exp(-x^2/20)
}

# Integral limits
a <- 0
b <- 1

#Bins
n <- 10 # increase this to see convergence
bins <- seq(a, b, length = n+1)

# Approximating the area under the curve
area.RS <- mean( fn(bins.low) * (b-a) )
area.RS
```

Estimating Riemann Integrals

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!