Stats 102C - Lecture 2-1: Monte Carlo Integration

Miles Chen, PhD

Week 2 Monday

Section 1

Lecture 2-1: Monte Carlo Integration

Review: Making predictions with the posterior distribution

On Friday, we ended lecture with the baseball example.

Our prior distribution for θ is a beta distribution with $\alpha = 81$ and $\beta = 219$. We observed a new player with 10 at bats and 5 hits. Our posterior distribution for θ is now a beta distribution with $\alpha = 86$ and $\beta = 224$.

"If this player has three at bats in the next game, what is the probability he gets exactly two hits?"

The answer depends on the value of θ: $\binom{3}{2}\theta^2(1-\theta)^1$

θ is a random variable.

We can estimate the expected value of the probability via Monte Carlo.

Estimate the expected value of the probability via Monte Carlo

We use Monte Carlo methods to estimate the expected value of $g(x)$

$$\mathbb{E}_f[g(X)] = \int_{\mathcal{X}} g(x) f(x) dx \approx \frac{1}{n} \sum_{j=1}^{n} g(x_j)$$

Where x_j are values randomly drawn from a distribution with PDF $= f(x)$.

$f(x)$ is the PDF of a beta distribution with $\alpha = 86$ and $\beta = 224$. We use R's `rbeta()` function to draw random values from this distribution.

The function $g(x)$ is the probability of getting 2 hits in 3 at bats: $\binom{3}{2}\theta^2(1-\theta)^1$

```
set.seed(1)
samp <- rbeta(10^6, 86, 224)
g <- function(theta){ 3 * theta ^ 2 * (1 - theta) }
mean(g(samp))
```

```
## [1] 0.1671577
```

Classical Monte Carlo Integration

$$\mathbb{E}_f[h(X)] = \int_{\mathcal{X}} h(x)f(x)dx \approx \frac{1}{n}\sum_{j=1}^{n} h(x_j)$$

$\mathbb{E}_f[h(X)]$: The expected value of the function $h()$ applied to the random variable X whose distribution is defined by the PDF $f()$

is equal to

$\displaystyle\int_{\mathcal{X}} h(x)f(x)dx$: the integral of the product of $h()$ and $f()$ across all values of x in the domain $\mathcal{X}$

which can be approximated by

$\displaystyle\frac{1}{n}\sum_{j=1}^{n} h(x_j)$ the mean of a sample of n values randomly drawn from the density f.

Side note: Notation

Random variable $X \sim f(x)$ for $x \in \mathcal{X}$

The region $\mathcal{X}$ is the *support* of X

- $f(x) > 0$ for $x \in \mathcal{X}$
- $f(x) = 0$ for $x \notin \mathcal{X}$
- $\displaystyle\int_{\mathcal{X}} f(x)dx = 1$

$\mathbb{E}_f[X]$ is the expectation of X a random variable with density function f.

Classical Monte Carlo Integration

Introduction to Monte Carlo Methods with R Section 3.2

If we are able to sample directly from the density function $f(x)$, we can estimate $\mathbb{E}_f[h(X)]$ as follows:

1. Generate $x_1, x_2, \ldots, x_n \overset{\text{iid}}{\sim} f(x)$.
2. Compute $h(x_1), h(x_2), \ldots, h(x_n)$.
3. Estimate $\mathbb{E}_f[h(X)]$ by

$$\bar{h}_n = \frac{1}{n} \sum_{i=j}^{n} h(x_j)$$

Monte Carlo Integration: How it works

Monte Carlo integration works because of the law of large numbers.

(not a formal proof)

As $n \to \infty$, we can expect the sample of random draws $\{x_1, x_2, \ldots, x_n\}$ to become more representative of the distribution made by the density function f.

That is to say, the empirical cumulative distribution function $(\hat{F}_n(t))$ based on the sample converges almost surely to the actual cumulative distribution function.

$$\hat{F}_n(t) \xrightarrow{a.s.} F(t)$$

As a result, $\bar{h}_n$ which is based on our sample will also converge to $\mathbb{E}_f[h(X)]$ as $n \to \infty$.

Section 2

What does it mean for the ECDF to converge to CDF?

Empirical CDF

(from wikipedia) Definition of empirical CDF:

Let $(X_1, \ldots, X_n)$ be independent, identically distributed random variables with the cumulative distribution function $F(t)$.

The ECDF is defined as:

$$\hat{F}_n(t) = \frac{\text{number of elements in the sample} < t}{n}$$

Compare the definition of the ECDF with the CDF:

$$F(t) = \Pr(X \leq t)$$

We can see that the ECDF is an empirical estimate of the CDF, where $\Pr(X \leq t)$ is approximated using the observations in the sample.

Distribution functions in R

In R, each probability distribution has 4 functions associated with it:

- `d...` is for density. The PDF or PMF of a probability distribution.
  - e.g. `dnorm(0)` = 0.3989423
- `p...` is for cumulative probability. The CDF of a probability distribution.
  - e.g. `pnorm(0)` = 0.5
- `q...` is for quantile. The inverse CDF of a probability distribution.
  - e.g. `qnorm(0.5)` = 0
- `r...` is for random. This generates a random draw from the probability distribution.

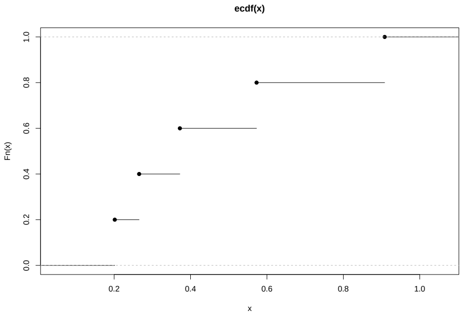
Example ECDF - Uniform

Let's look at the ECDF for five random values drawn from a uniform distribution.

```
set.seed(1)
x <- runif(5)
sort(x)
```

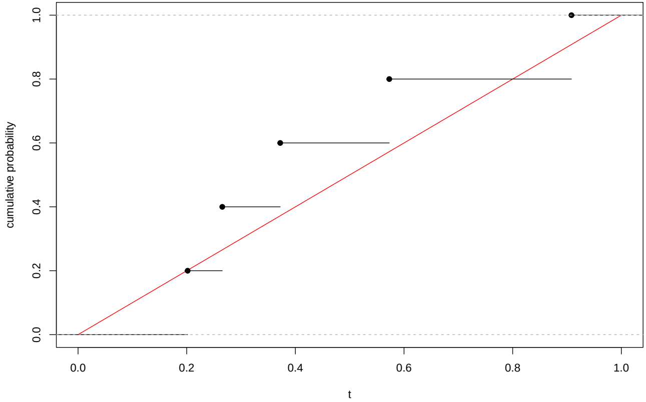
```
## [1] 0.2016819 0.2655087 0.3721239 0.5728534 0.9082078
```

```
plot(ecdf(x))
```



ECDF vs CDF as $n \to \infty$ - Uniform

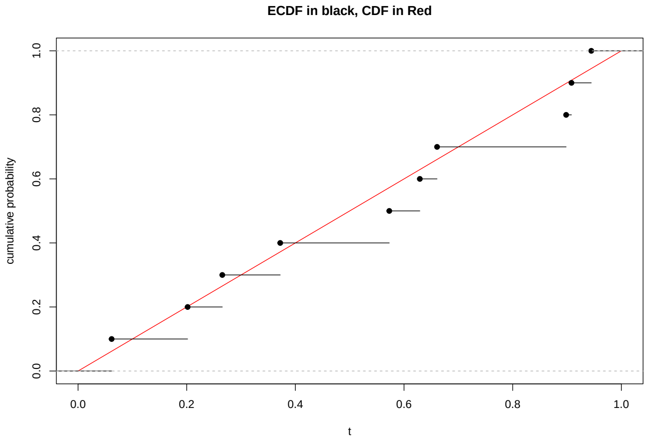
```r
t <- seq(0,1, by = .01)
plot(t, punif(t), type = "l", ylab = "cumulative probability", col = "red") # Uniform CDF
plot(ecdf(x), add = TRUE) # ECDF based on our 5 observations
```



ECDF vs CDF as $n \to \infty$ - Uniform

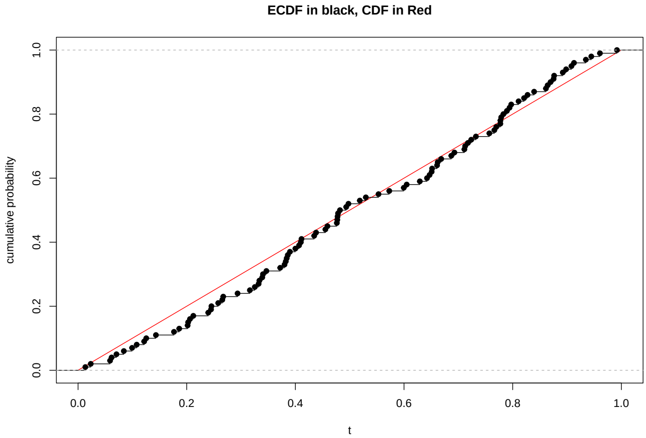
```r
n <- 10
```

```r
plot(t, punif(t), type = "l", ylab = "cumulative probability", col = "red", main = "ECDF in black, CDF in Red") # Uniform CDF
set.seed(1)
x <- runif(n)
plot(ecdf(x), add = TRUE)  # ECDF based on n observations
```



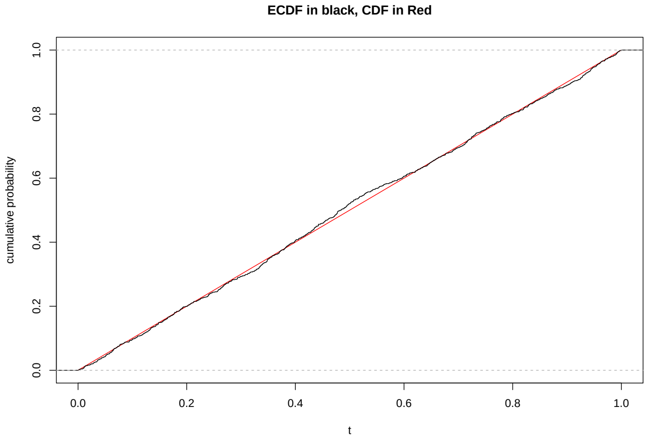
ECDF vs CDF as $n \to \infty$ - Uniform

```
n <- 100
```



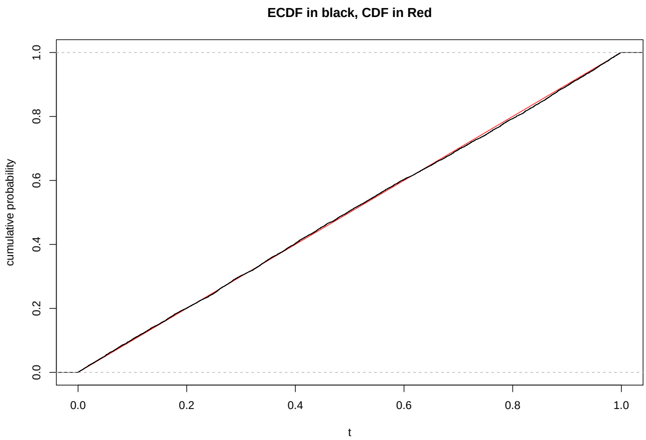
ECDF vs CDF as $n \to \infty$ - Uniform

```r
n <- 1000
```



ECDF vs CDF as $n \to \infty$ - Uniform

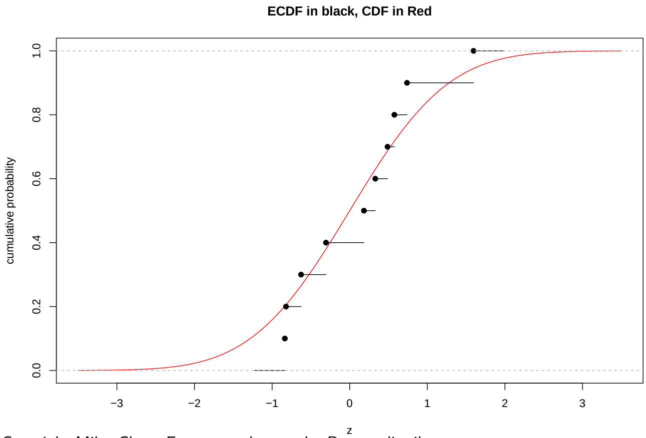
```r
n <- 10^4
```



ECDF vs CDF as $n \to \infty$ - Normal

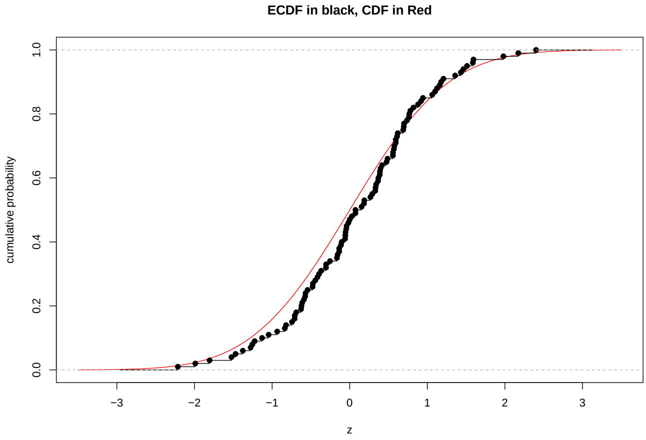
```
n <- 10; z <- seq(-3.5,3.5, by = .01)
```

```
plot(z, pnorm(z), type = "l", ylab = "cumulative probability", col = "red", main = "ECDF in black, CDF in Red") # normal CDF
set.seed(1)
x <- rnorm(n)
plot(ecdf(x), add = TRUE) # ECDF
```



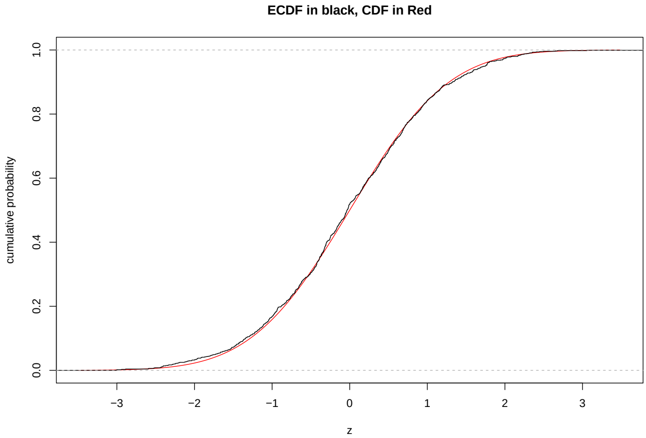
ECDF vs CDF as $n \to \infty$ - Normal

```
n <- 100
```



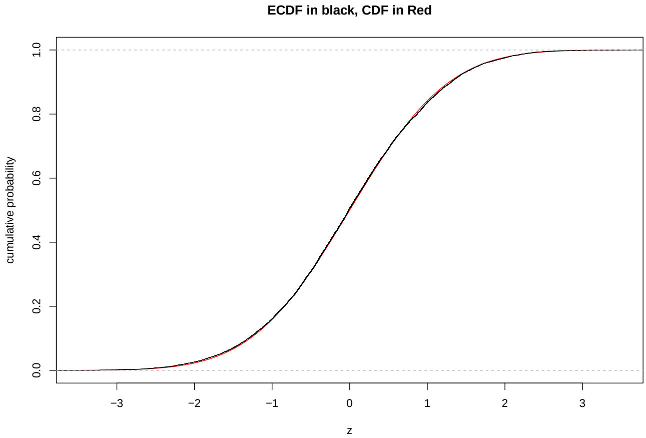
ECDF vs CDF as $n \to \infty$ - Normal

```r
n <- 1000
```



ECDF vs CDF as $n \to \infty$ - Normal

```r
n <- 10^4
```



$\hat{F}_n(t) \longrightarrow F(t)$

As n gets larger, we can see that the empirical cumulative distribution function $(\hat{F}_n(t))$ based on the sample gets closer and closer to the actual cumulative distribution function.

While the previous diagrams do not constitute a proof, they do provide some intuition behind the idea of using a large sample of random values to approximate a function applied to a random variable.

Section 3

Properties of the Monte Carlo Estimator $\bar{h}_n$

Expected value of the estimate $\bar{h}_n$

As shown below, $\bar{h}_n$ is an unbiased estimator of $\mathbb{E}[h(X)]$

$$
\begin{aligned}
\mathbb{E}[\bar{h}_n] &= \mathbb{E}\left[\frac{1}{n}\sum_{i=j}^{n} h(X_j)\right] \\
&= \frac{1}{n}\sum_{i=j}^{n} \mathbb{E}\left[h(X_j)\right] \\
&= \mathbb{E}\left[h(X)\right]
\end{aligned}
$$

Variance of the estimate $\bar{h}_n$

$\bar{h}_n$ is the Monte Carlo Estimator of $\mathbb{E}_f[h(X)]$

$$\bar{h}_n = \frac{1}{n}\sum_{i=j}^{n} h(x_j) \approx \mathbb{E}_f[h(X)] = \int_{\mathcal{X}} h(x)f(x)dx$$

The variance of this estimator is then

$$\begin{aligned}
Var(\bar{h}_n) &= Var\left(\frac{1}{n}\sum_{i=j}^{n} h(X_j)\right) \\
&= Var\left(\frac{1}{n}h(X_1) + \frac{1}{n}h(X_2) + \cdots + \frac{1}{n}h(X_n)\right) \\
&= \frac{1}{n^2}Var(h(X_1)) + \frac{1}{n^2}Var(h(X_2)) + \cdots + \frac{1}{n^2}Var(h(X_n))
\end{aligned}$$

Because all of the X_j are i.i.d., they have the same variance. All of $h(X_j)$ will also have the same variance.

$$Var(\bar{h}_n) = \frac{1}{n^2}(n \cdot Var(h(X))) = \frac{1}{n} \cdot Var(h(X))$$

Monte Carlo estimate of the Variance of the estimate $\bar{h}_n$

If we do not know the variance of $h(X)$ directly, we can also estimate it using some identities and Monte Carlo methods. The definition of the variance for a continuous variable is $Var(Y) = \int_{\mathcal{Y}} (y - \mathbb{E}[Y])^2 f(y) dy$. If we let $Y = h(X)$, we have:

$$Var(\bar{h}_n) = \frac{1}{n} \cdot Var(h(X)) = \frac{1}{n} \int_{\mathcal{X}} (h(x) - \mathbb{E}_f[h(X)])^2 f(x) dx$$

If we let $g(x) = (h(x) - \mathbb{E}[h(X)])^2$, we know the integral can be approximated using the Monte Carlo method by drawing a large sample of x_j values from the density f. $\int_{\mathcal{X}} g(x) f(x) dx \approx \frac{1}{n} \sum_{j=1}^{n} g(x_j)$

$$Var(\bar{h}_n) \approx \frac{1}{n} \cdot \frac{1}{n} \sum_{j=1}^{n} g(x_j) = \frac{1}{n} \cdot \frac{1}{n} \sum_{j=1}^{n} (h(x_j) - \mathbb{E}_f[h(X)])^2$$

Monte Carlo estimate of the Variance of the estimate $\bar{h}_n$

$$Var(\bar{h}_n) \approx \frac{1}{n} \cdot \frac{1}{n} \sum_{j=1}^{n} (h(x_j) - \mathbb{E}_f[h(X)])^2$$

Finally, we replace $\mathbb{E}[h(X)]$ with our Monte Carlo Estimate $\bar{h}_n$. With this, we define v_n as the Monte Carlo estimate of the variance of the Monte Carlo estimate $\bar{h}_n$

$$Var(\bar{h}_n) \approx v_n := \frac{1}{n^2} \sum_{j=1}^{n} (h(x_j) - \bar{h}_n)^2$$

Central Limit Theorem

Because $\bar{h}_n$ is effectively a sample mean of $h(X)$, we can use the central limit theorem to approximate its sampling distribution.

The Central Limit Theorem states that if $\bar{y}$ is a sample mean, its sampling distribution can be approximated with a normal distribution with mean $= \mu$ and variance $= \frac{1}{n}s^2$, where $\mu = \mathbb{E}[Y]$ and $s^2 = Var(Y)$

$$\bar{h}_n \sim \mathcal{N}\left(\mathbb{E}_f[h(X)], \frac{1}{n}Var(h(X))\right)$$

We showed $\frac{1}{n}Var(h(X)) \approx v_n$ which we found in the previous slides.

$$\bar{h}_n \sim \mathcal{N}\left(\mathbb{E}_f[h(X)], v_n\right)$$

Knowing that $\bar{h}_n$ is normally distributed allows us to create confidence bounds.

Recap

Let's not get lost with all the math.

The important things so far are this:

- If we can sample values of X from f, we have a way to estimate the expected value of a function $h(x)$ when applied to a random variable X.
  - $\mathbb{E}_f[h(X)] \approx \bar{h}_n$
  - $\bar{h}_n$ is the sample mean of $h()$ applied to random draws of X.
  - It works because of the Law of Large numbers (we saw ECDF approach CDF)
- This estimator is $\bar{h}_n$ normally distributed (because of Central Limit Theorem)
  - We can estimate the variance of the estimator using v_n

Section 4

Applications of Monte Carlo Integration

Estimate an integral of any function

Let's say you want to estimate the integral of a function $h(x)$ on a closed interval (a, b):

$$I = \int_a^b h(x)dx$$

Perhaps $h(x)$ is difficult to integrate directly.

Estimate an integral of any function

The average value of $h(x)$ on the interval (a, b) is

$$\frac{1}{b-a}\int_a^b h(x)dx$$

We can rewrite this average as:

$$\frac{1}{b-a}\int_a^b h(x)dx = \int_a^b h(x)\frac{1}{b-a}dx = \int_a^b h(x)f(x)dx = \mathbb{E}_f[h(X)] \approx \bar{h}_n$$

Where $f(x) = \frac{1}{b-a}$, which is the PDF of a uniform distribution on the interval (a,b).

The value of the original integral is then:

$$I = \int_a^b h(x)dx \approx (b-a)\bar{h}_n$$

Estimate an integral of any function

We can estimate $\mathbb{E}_f[h(X)]$ as follows:

1. Generate $x_1, x_2, \ldots, x_n \overset{\text{iid}}{\sim} Unif(a, b)$.
2. Compute $h(x_1), h(x_2), \ldots, h(x_n)$.
3. Estimate $\mathbb{E}_f[h(X)]$ by

$$\bar{h}_n = \frac{1}{n} \sum_{i=j}^{n} h(x_j)$$

4. Estimate I with

$$\hat{I} = (b - a)\bar{h}_n = \frac{(b - a)}{n} \sum_{i=j}^{n} h(x_j)$$

Example

$$h(x) = [\cos(50x) + \sin(20x)]^2$$

Estimate the following integral via Monte Carlo integration:

$$\int_0^1 h(x)dx = \int_0^1 [\cos(50x) + \sin(20x)]^2 dx$$

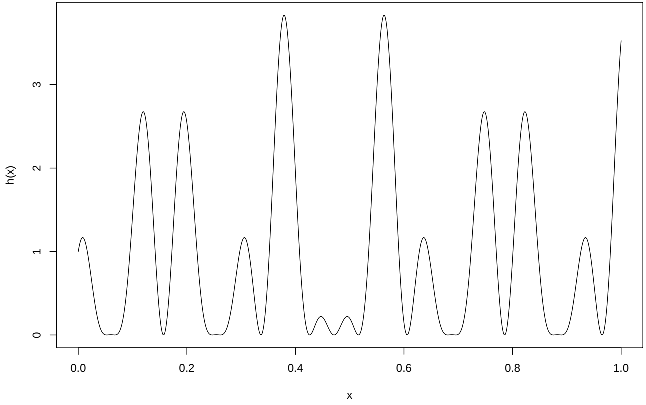
Wolfram alpha

https://www.wolframalpha.com/input/?i=%5Cint_0%5E1+%5B%5Ccos%2850x%29+%2B+%5Csin%2820x%29%5D%5E2+dx

Wolfram Alpha says the answer is

$$1 + \cos(30)/30 - \cos(70)/70 - \sin(40)/80 + \sin(100)/200 - 2/105 \approx 0.96520$$

```r
x <- seq(0, 1, by = .001)
h <- function(x) {(cos(50 * x) + sin(20 * x)) ^ 2}
plot(x, h(x), type = "l")
```



Estimate via Monte Carlo

```r
set.seed(1)
n <- 10^5 # Specify the number of points to generate
# Generate n points from Unif(0,1)
X <- runif(n, 0, 1)
# Compute h(X)
h_X <- h(X)
# Compute mean(h(X))
mean(h_X)
```

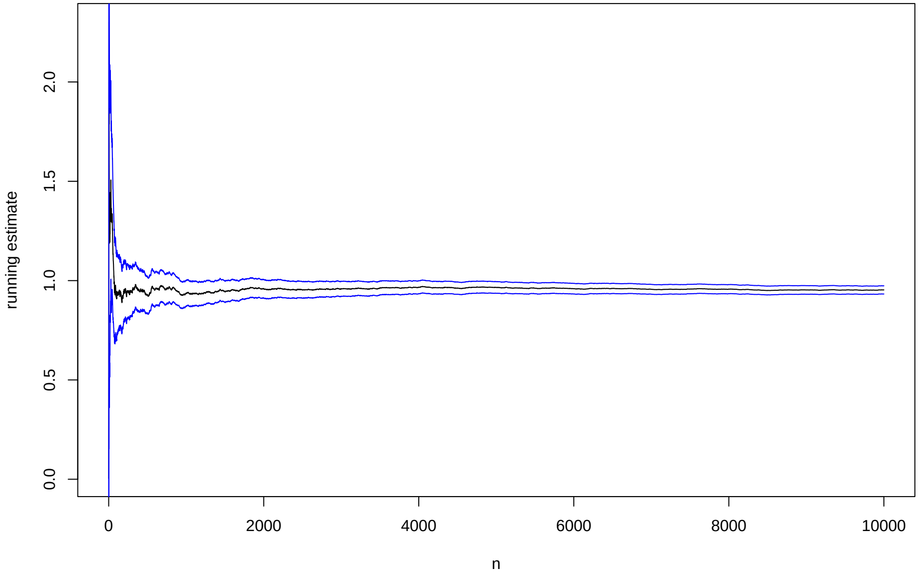
```
## [1] 0.966863
```

$$\bar{h}_n = 0.966863$$

$$\hat{I} = (b-a)\bar{h}_n = (1-0)\bar{h}_n = \bar{h}_n = 0.966863$$

```r
n <- 10^4 # graph just the first 10000 values

# Compute cumulative mean(h(X))
hbar_n <- cumsum(h_X[1:n])/(1:n)

# Function to estimate Var(hbar_n) (see slide 27)
var_m <- function(m) {
# Estimate Var(hbar_m) for any given m
sum((h_X[1:m] - hbar_n[m]) ^ 2) / m ^ 2
}

# running estimates of the variance
v_n <- apply(t(1:n), 2, var_m)

# Compute standard error
se_n <- sqrt(v_n)
```

```r
# Plot cumulative mean against iterations
plot(1:n, hbar_n, type = "l" , xlab = "n", ylab = "running estimate")

# Add approximate 95% confidence band
lines(hbar_n + 1.96 * se_n, col = "blue")
lines(hbar_n - 1.96 * se_n, col = "blue")
```


running estimate
0.0
0.5
1.0
1.5
2.0
0
2000
4000
6000
8000
10000
n

