Introduction to Monte Carlo

Computation Summer Student Seminar Series

Mike Pozulp
Computer Scientist

June 21, 2017

LLNL-PRES-734172

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Lawrence Livermore National Laboratory

About Me

- Grew up in western suburbs of Chicago
- Graduated in 2015 from the College of William & Mary in Virginia with B.S. in C.S.
- Joined LLNL in July 2015 and assigned to Monte Carlo radiation transport code team
- Interests: Monte Carlo, compilers, physics
  - I get to use all three at LLNL

Me as a summer student

Introduction

- Monte Carlo is a numerical method
- Monte Carlo can be used to:
    1. Support analytic solutions
    2. Solve problems without analytic solutions
- R is a powerful programming language well-suited to Monte Carlo simulation
- At LLNL, we use Monte Carlo to do 1. and 2.

Outline

- History of Monte Carlo
- Mathematical underpinnings
  - Random variables
  - Probability functions
  - Random variate generation
- Examples
  - MC Integration
  - Point Estimator Probability

History of Monte Carlo



- Ulam, von Neumann, and Metropolis developed Monte Carlo in the 1940's
  - All were employees at Los Alamos National Lab
- Named after Monte Carlo Casino in Monaco

Image credits: (top) Los Alamos National Lab (bottom) https://www.lancaster.ac.uk/pg/jamest/Group/intro2.html

- Initially used for neutron transport
- Used today in biology, finance, engineering, computer graphics, …

Random Variables

- A **random variable** is a function X that maps each outcome $s \in S$ in the **sample space** of a **random experiment** to one real number $X(s) = x$ in the **support** $\mathcal{A} = \{x | x = X(s), s \in S\}$

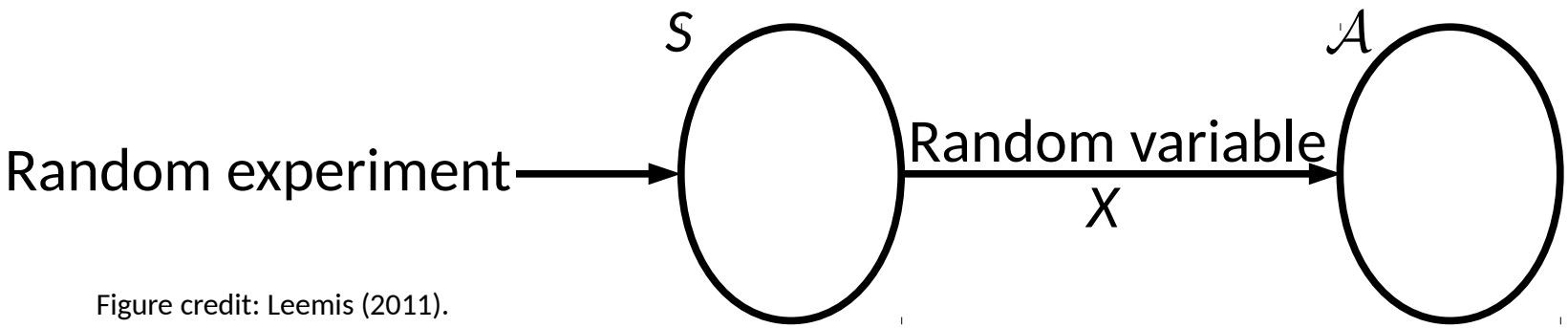


Figure credit: Leemis (2011).

- Consider the **random experiment** in which two coins are flipped
  - S = {HH, HT, TT, TH}, E = {HT}, P(E) = ¼
  - Define X = number of heads appearing in the two tosses
  - $\mathcal{A} = \{ x | x = 0, 1, 2\}$, $A = \{x | x = 2\}$, $P(X \in A) = ¼$

Probability Functions

- Probability mass function (pmf)
  - $f(x) = P(X = x)$ is the probability that X takes on the value x as the result of a random experiment. $f(x)$ satisfies

$$\sum_{\mathcal{A}} f(x) = 1 \qquad f(x) > 0, x \in \mathcal{A}$$

- Probability density function (pdf)
  - continuous analog of pmf

$$\int_{\mathcal{A}} f(x)dx = 1 \quad f(x) \geq 0, x \in \mathcal{R}$$

- Cumulative distribution function (cdf)
  - $F(x) = P(X \leq x)$ is the probability that X takes on a value less than or eq to x

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(w)dw$$

  - Since $F(x)$ is a probability, $0 \leq F(x) \leq 1$

Random Variate Generation

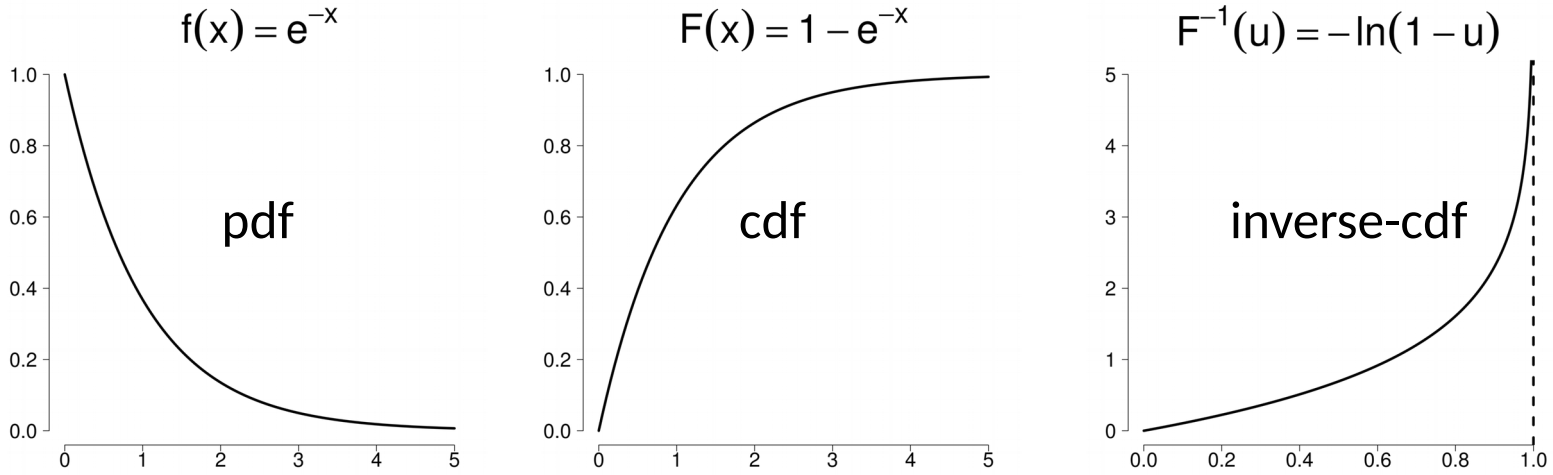
- An instance of a random variable *X*, denoted *x* and called a **random variate**, can be generated via

$$x \leftarrow F^{-1}(u)$$

where the inverse cdf F^{-1} exists and random variate $u \sim U(0,1)$

  - Every computer comes with *U*(0,1) PRNG

- Consider an exponentially-distributed random variable *X*



Example 1 MC Integration

- Find $I = \int_{-\infty}^{\infty} e^{-x^2} dx$
- Not straightforward analytically. The trick is to find I^2 instead

$$
\begin{aligned}
I^2 &= \int_{-\infty}^{\infty} e^{-x^2} dx \int_{-\infty}^{\infty} e^{-x^2} dx \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-(x^2+y^2)} dx\, dy
\end{aligned}
$$

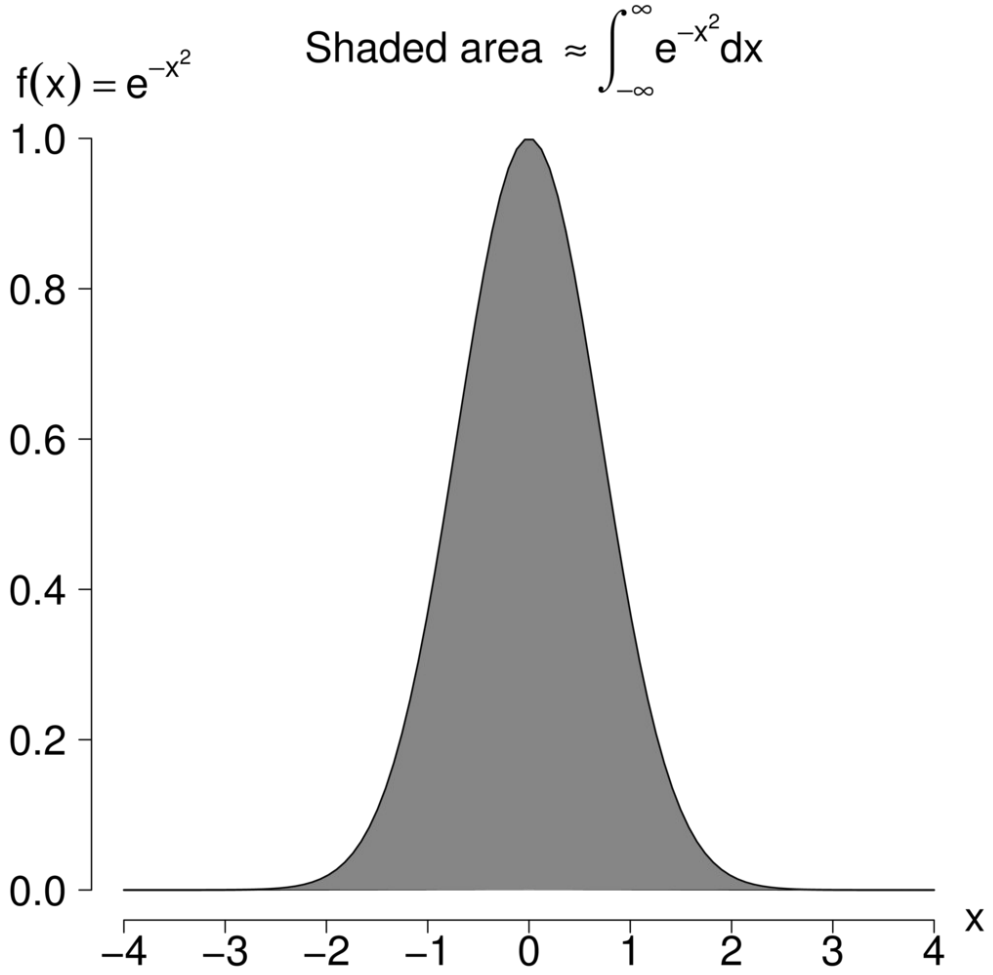
switching to polar coordinates

$$
\begin{aligned}
I^2 &= \int\int_{\mathcal{R}^2} e^{-(x^2+y^2)} d(x,y) \\
&= \int_0^{2\pi} \int_0^{\infty} e^{-r^2} r\, dr\, d\theta \\
&= \pi
\end{aligned}
$$

thus $\boxed{I = \sqrt{\pi}}$

Example 1 MC Integration

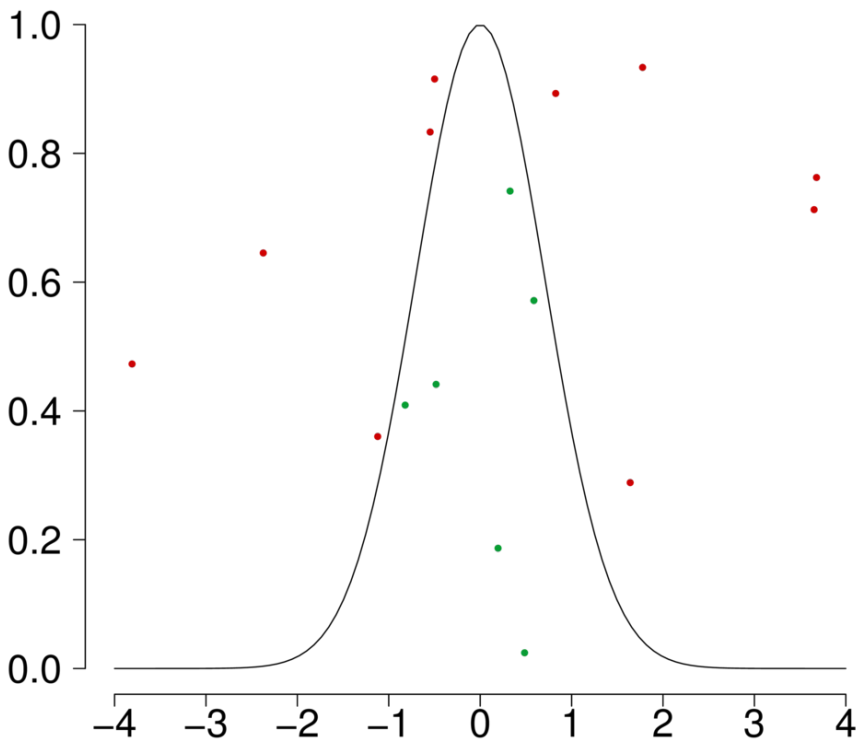
- Consider the same problem geometrically: the area under the curve is the integral
- Is there a way to calculate the shaded area numerically?
  - **Monte Carlo integration**



Describing MC Integration

- Imagine throwing darts at the image on the right
- We can find the integral by

1. Counting the number of darts landing inside the curve
2. Dividing by the number of darts thrown
3. Multiplying by the total area



The R Programming Language

- R is an open source programming language
- R first appeared in 1993
- The R interpreter
  - scripting, data exploration, graphics
- R package library
  - huge community of active users
- R language features
  - Vector arithmetic, functional programming support, slow imperative code execution, {r,p,d,q}$dist

Image credit: The R Foundation (CC-BY-SA 4.0).
https://www.r-project.org/logo/

Example 1 MC Integration

- Using R to find $I = \int_{-\infty}^{\infty} e^{-x^2} dx$ by Monte Carlo integration

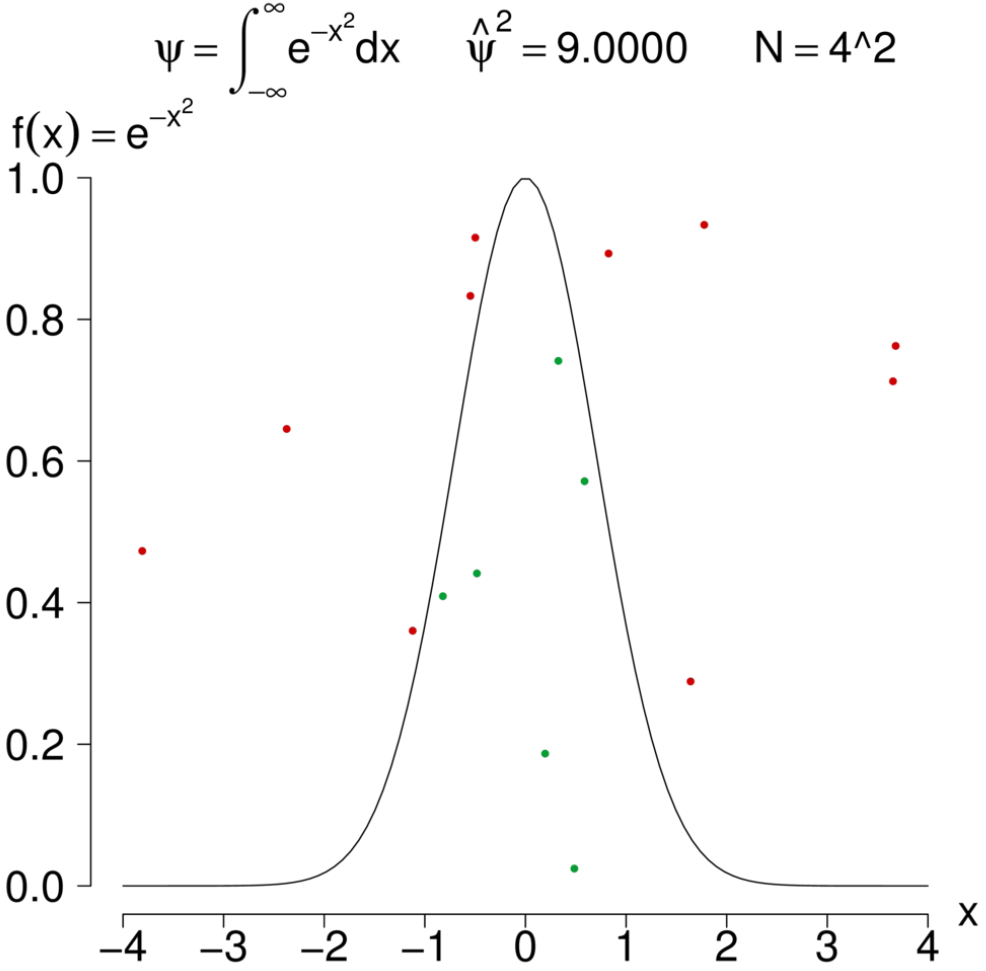
```
 1 N <- 4 ** 2                # number of replications
 2 x <- runif(N, -4, 4)       # randomly sample 8x1 rect centered at 0 by generating
 3 y <- runif(N, 0, 1)        # N random variates (x,y) where x ~ U(-4,4), y ~ U(0,1)
 4
 5 fx <- exp(-(x*x))          # evaluate function to find position on curve
 6 inside <- y < fx           # if y < f(x) the sampled point is inside the curve
 7
 8 # scale inside/total by the total area to get the estimator for the integral
 9 inside_ratio <- sum(inside) / length(inside)
10 total_area <- 8
11 integral_estimator <- inside_ratio * total_area
12
13 # estimator^2 should converge to pi (recall our analytic solution was sqrt(pi))
14 pi_estimate <- integral_estimate ** 2
15 print(pi_estimate)
```

- Running for N = 4^1, 4^2, …, 4^12 yields e.g.

```
[0] 0.000, 6.250, 4.516, 2.066, 3.314, 3.443, 3.111, 3.094, 3.144, 3.136, 3.141, 3.140
```

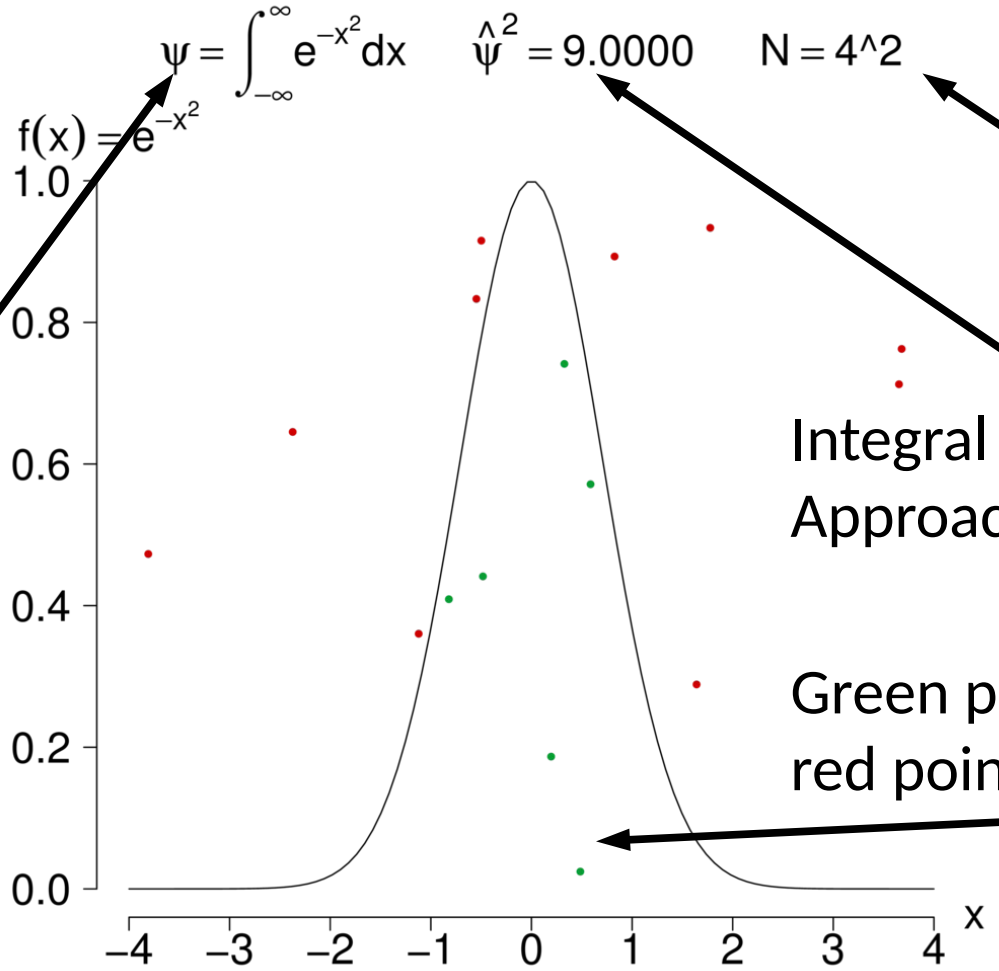
Example 1 MC Integration

- Graphically, we see



Example 1 MC Integration

- Graphically, we see



Sample size

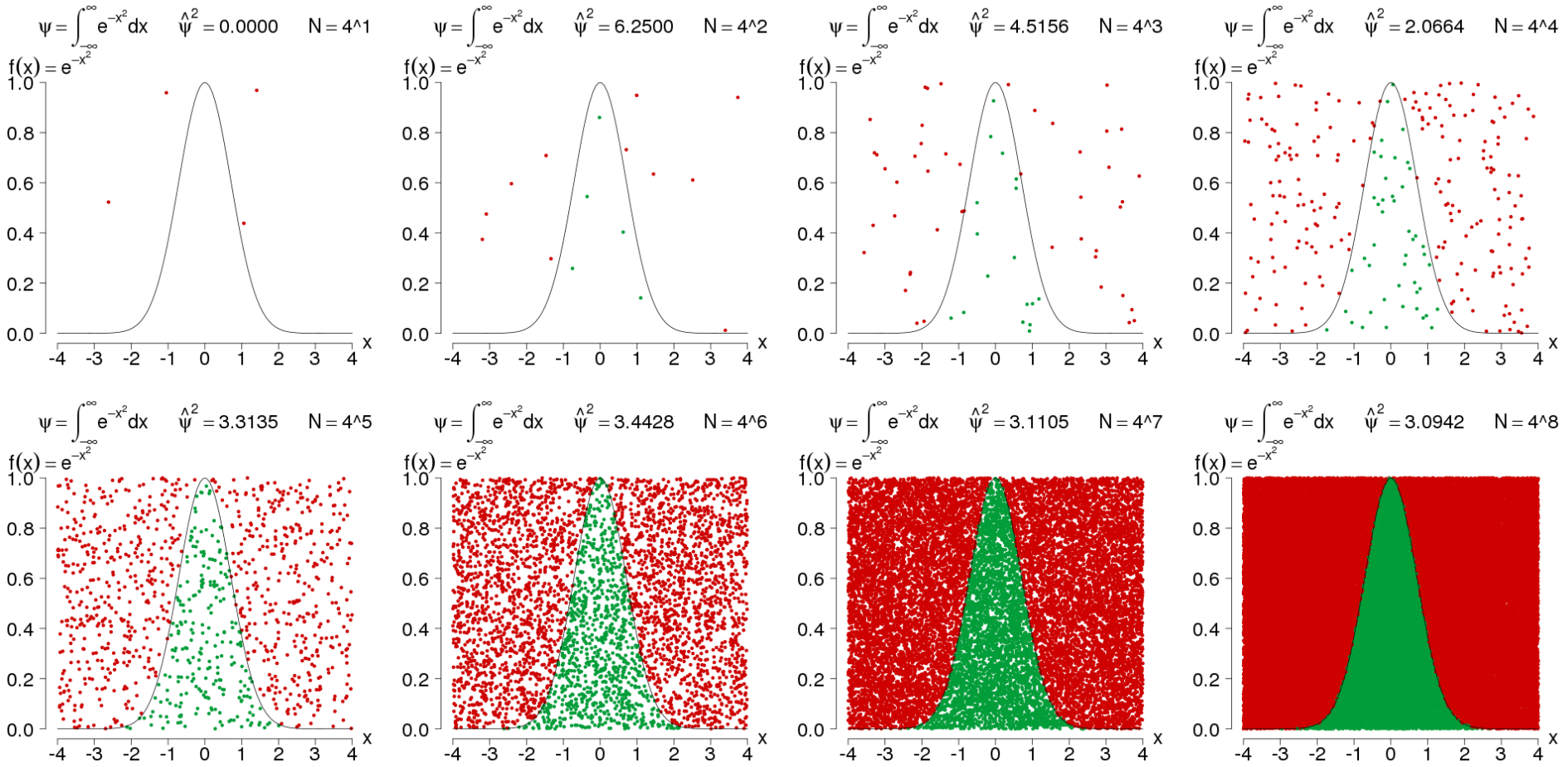
The integral that we are evaluating

Integral estimator squared. Approaches π as N ⟶∞

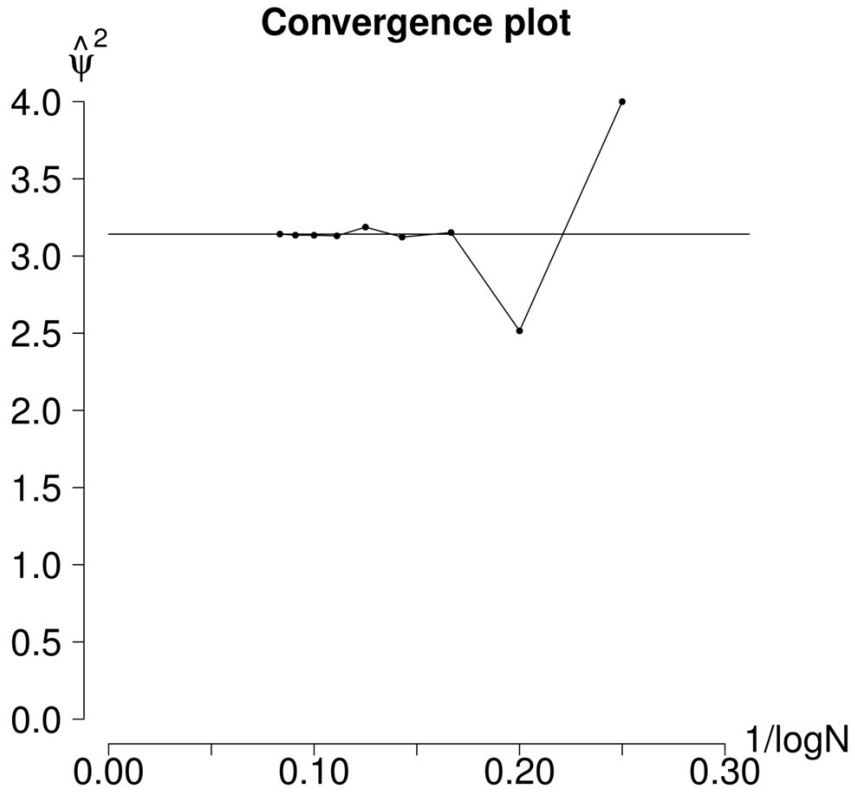
Green points are inside and red points are outside curve

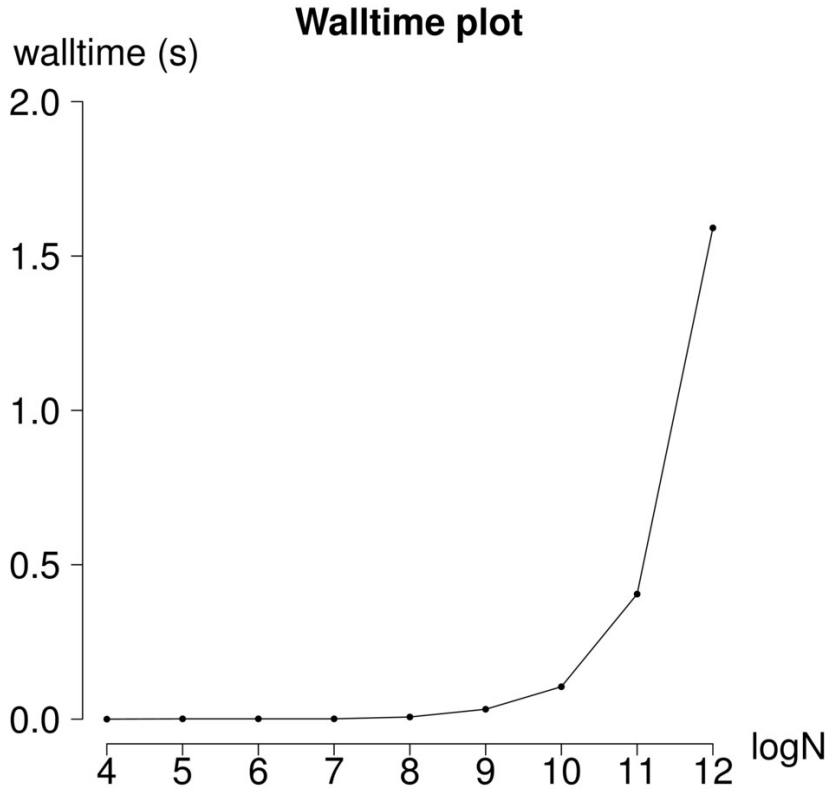
Example 1 MC Integration

- Visual convergence: colored dots appear as solid fill when $N \to \infty$



Example 1 MC Integration





- $\hat{\Psi}^2 \rightarrow \pi$ as $N \rightarrow \infty$ in support of our analytic solution
- Increasing *N* gives us a better estimator, but also increases runtime

MC Error Analysis

- The MC error is proportional to

$$\frac{s}{\sqrt{N}}$$

  where s is the **sample standard deviation** and *N* is the **sample size**

- To acheive 2x error reduction we need 4x the sample size

- (Figure A) The curve with **population standard deviation** σ = 2 has greater **spread** than σ = 1

- (Figure B) Bumps along edges are **noise**, a commonly used synonym for error

Figure A

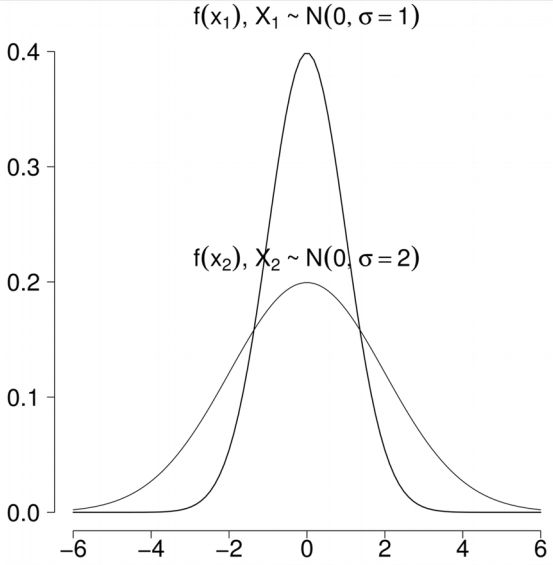
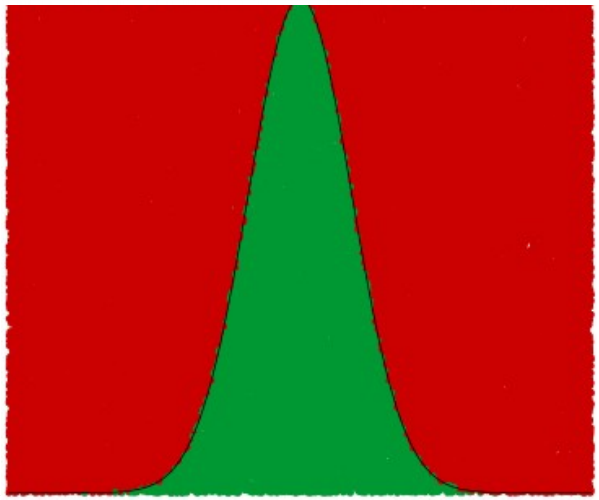


Figure B



Example 2 Point Estimator Probability

Let X_1 and X_2 be a random sample from a $U(0, \theta)$ population, where θ is an unknown positive parameter. The three point estimators

- $2\bar{X}$,
- $3X_{(2)}/2$,
- $3X_{(1)}$,

can be used to estimate θ.

1. What are the probabilities, stated as exact fractions, that each of the three estimators is closest to the true value of θ when $\theta = 10$?

Problem credit: Leemis (2017). Personal communication.

- $\bar{X} = \frac{X_1+X_2}{2}$ is the **sample mean**
- $X_{(1)} = \min(X_1, X_2)$ and $X_{(2)} = \max(X_1, X_2)$ are **order statistics**

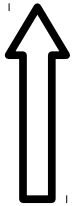
Example 2 Point Estimator Probability

Let X_1 and X_2 be a random sample from a $U(0, \theta)$ population, where θ is an unknown positive parameter. The three point estimators

- $2\bar{X}$,
- $3X_{(2)}/2$,
- $3X_{(1)}$,

can be used to estimate θ.

there exists an analytic solution



1. What are the probabilities, stated as exact fractions, that each of the three estimators is closest to the true value of θ when $\theta = 10$?

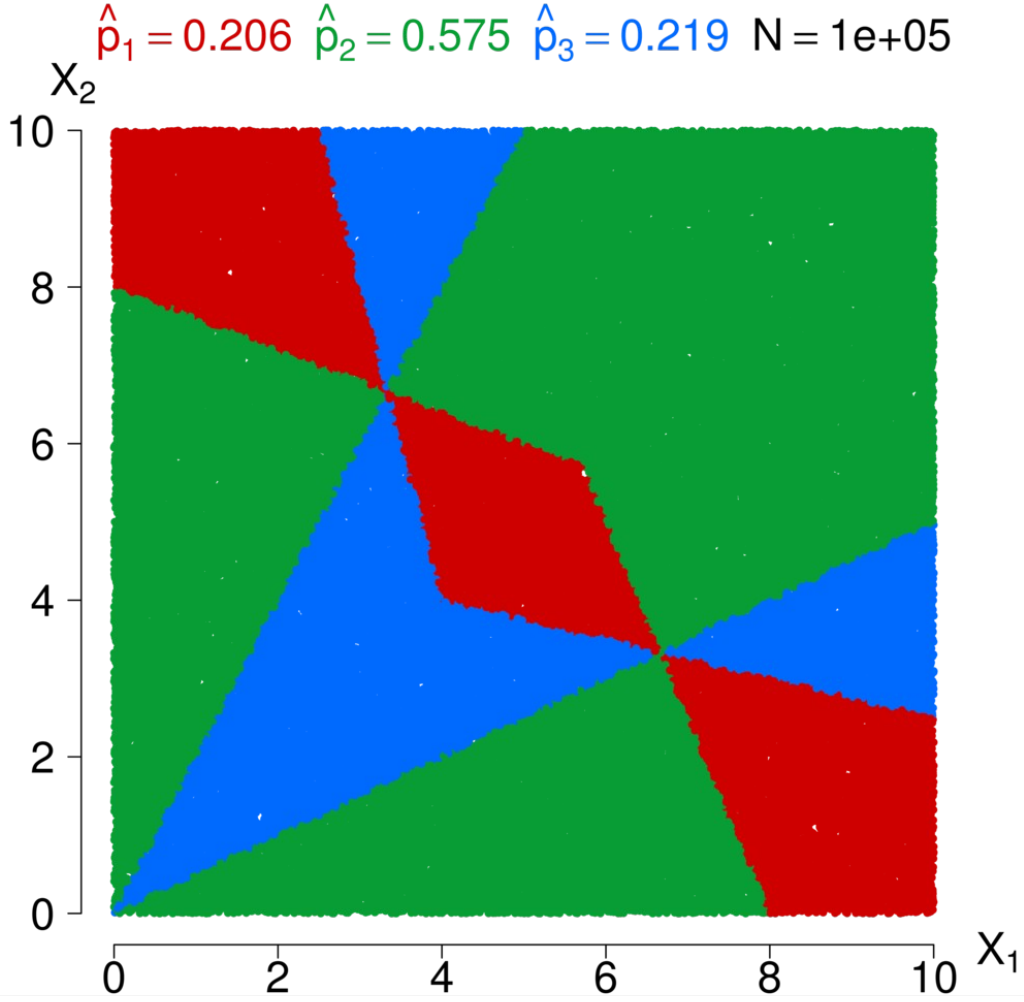
- $\bar{X} = \frac{X_1+X_2}{2}$ is the **sample mean**
- $X_{(1)} = \min(X_1, X_2)$ and $X_{(2)} = \max(X_1, X_2)$ are **order statistics**

Example 2 Point Estimator Probability

```r
N <- 10 ** 5                    # number of replications
theta <- 10                     # from problem description

x1 <- runif(N, 0, theta)        # randomly sample 10x10 rect by generating
x2 <- runif(N, 0, theta)        # N random samples (x1, x2) ~ (U(0,10),U(0,10))

xmat <- matrix(c(x1, x2), ncol = 2)
xbar <- apply(xmat, 1, mean)# Create Nx2 matrix and use apply() to compute
xmax <- apply(xmat, 1, max) # entry-by-entry {mean,min,max}(x1, x2)
xmin <- apply(xmat, 1, min) # where each pair (x1, x2) is one sample

est1 <- 2 * xbar                # Use sample statistics to compute the
est2 <- 3 * xmax / 2            # three point estimators given in the
est3 <- 3 * xmin                # problem description

diff1 = abs(est1 - theta)       # Winning estimator is the one which is closest
diff2 = abs(est2 - theta)       # to the parameter value. Create Nx3 matrix and
diff3 = abs(est3 - theta)       # use apply() to get winner for each sample.

diffmat <- matrix(c(diff1, diff2, diff3), ncol = 3)
closest <- apply(diffmat, 1, which.min)

p1hat <- sum(closest == 1) / length(closest) # Quotient wins/total is the MC
p2hat <- sum(closest == 2) / length(closest) # estimator for the prob that
p3hat <- sum(closest == 3) / length(closest) # estimator is closest to theta
```

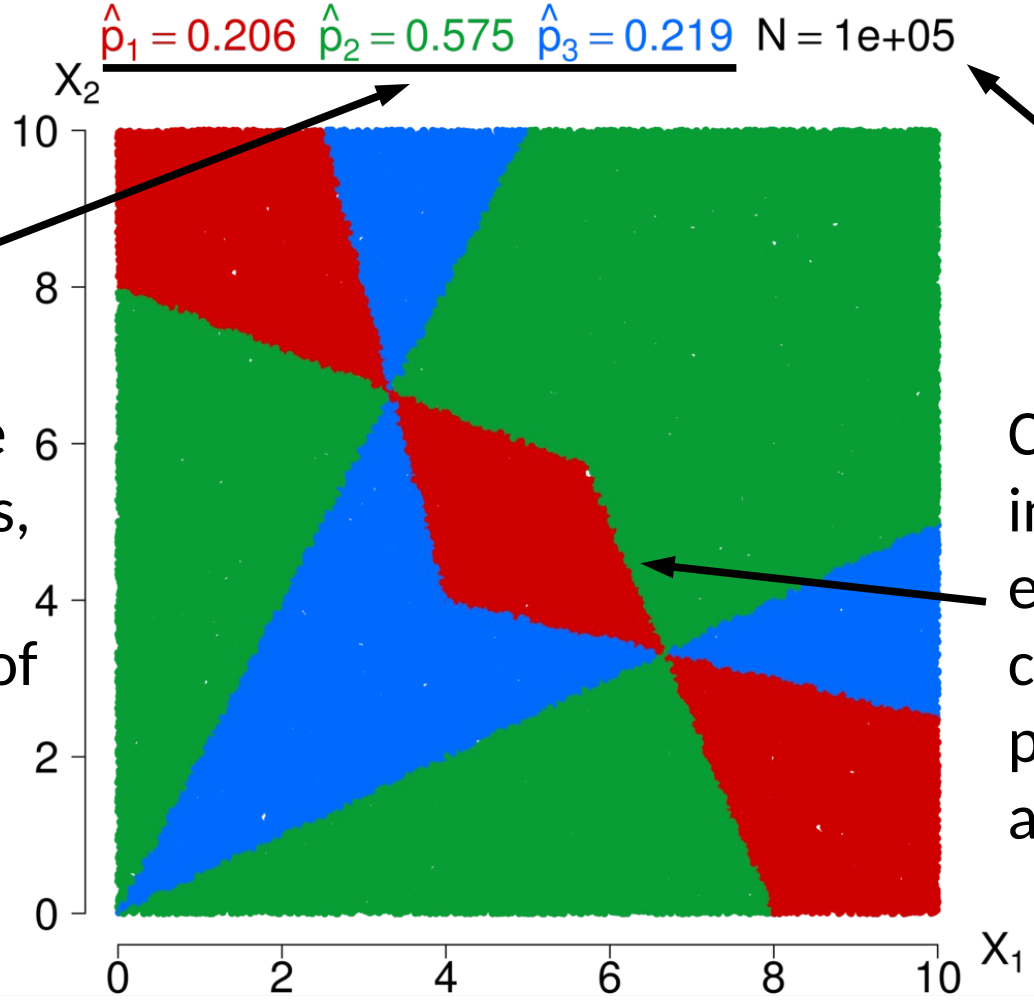
Example 2 Point Estimator Probability

- Graphically, we see



Example 2 Point Estimator Probability

- Graphically, we see



Sample size

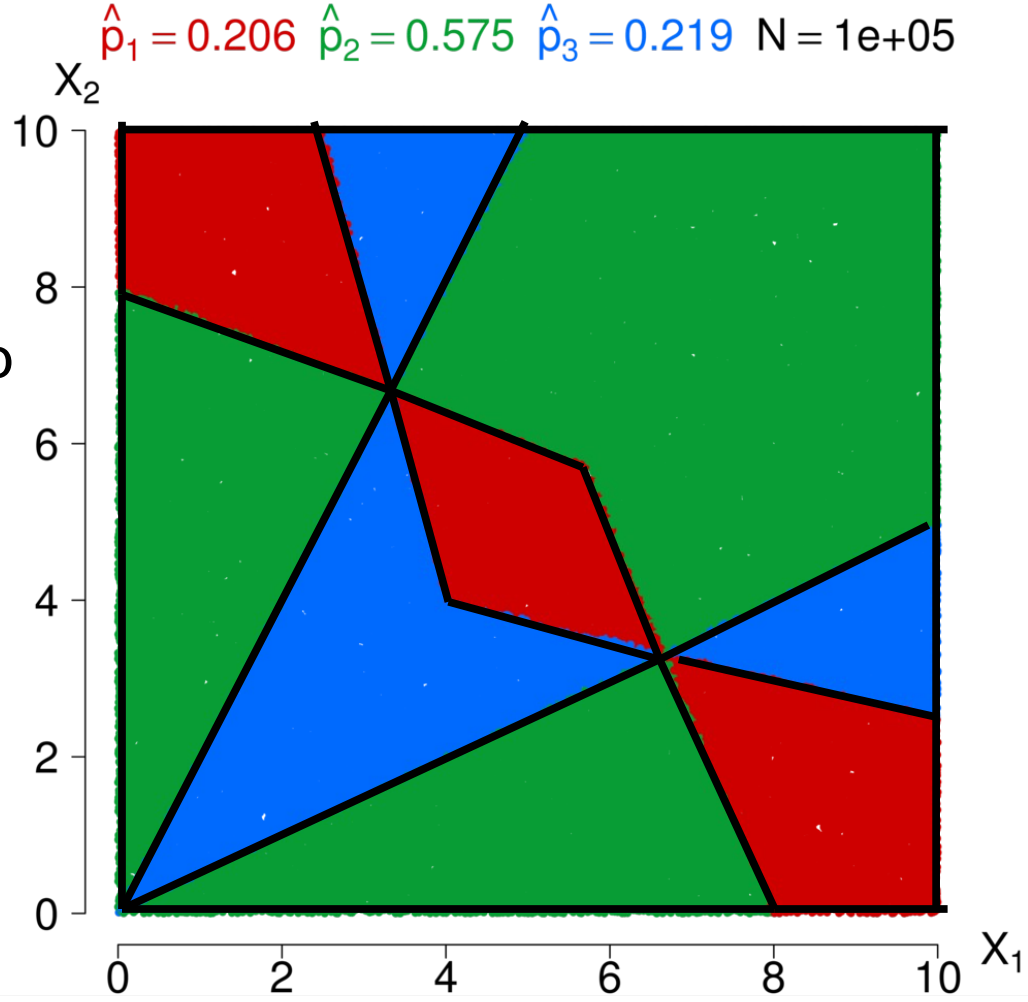
The probability estimator for the 3 point estimates, which are equivalent to % of plot covered by each color

Color of (x1, x2) indicates which estimator is closest to parameter θ = 10 at (x1, x2)

Example 2 Point Estimator Probability

- Graphically, we see

Use a ruler and area formulas to find analytic solution



Summary

- Monte Carlo is a numerical method
- Monte Carlo can be used to:
    1. Support analytic solutions
    2. Solve problems without analytic solutions
- R is a powerful programming language well-suited to Monte Carlo simulation
- At LLNL, we use Monte Carlo to do 1. and 2.

Recognition

- Thanks to Kate Burnett, Marcey Kelley, Tom Stitt, and Bujar Tagani for organizing the seminar series, inviting me to present, and recommending the topic of this talk

- Thanks to Jason Burnstein, Katie Schmidt, and Maren Hunsberger for providing valuable feedback on an early version of this talk

- Thanks to Professor Leemis for inspiring this talk, teaching me all of its contents, and continuing to inspire me

References

- Leemis, 2011. *Probability*.
- Leemis, 2016. *Learning Base R*.
  - The plot styling that I use comes from “Chapter 21. Custom Graphics” p. 152-154
- Leemis, 2017. Personal communication.
- Web resources accessed on June 20, 2017
  - http://www.lanl.gov/about/history-innovation/badges.php
  - https://www.lancaster.ac.uk/pg/jamest/Group/intro2.html
  - https://en.wikipedia.org/wiki/R_(programming_language)
  - https://en.wikipedia.org/wiki/Gaussian_integral

Lawrence Livermore
National Laboratory