Monte Carlo Methods in Inference

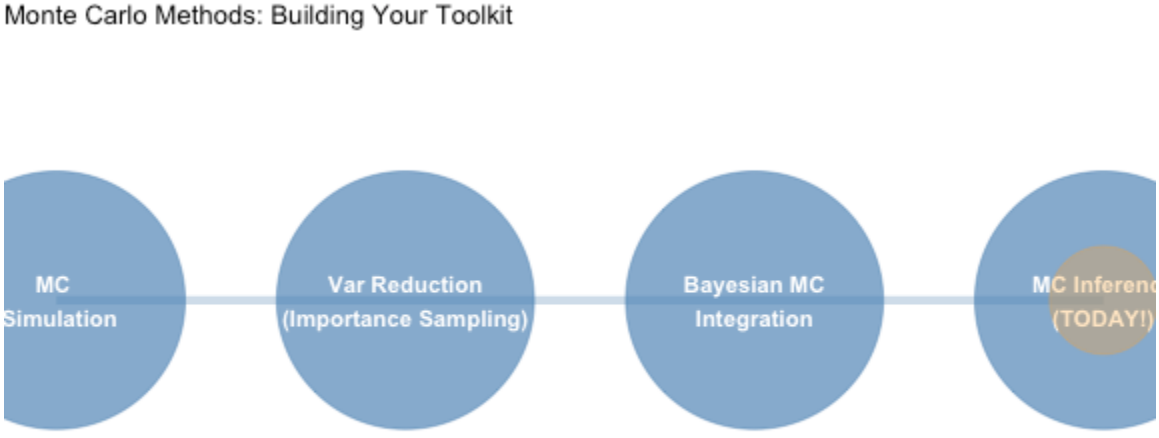
PSTAT 194CS

Dr. Uma Ravat

2026-01-28

Where We've Been & Where We

Course Journey: MC Methods To



Previously:

- Generating random samples
- Estimating integrals
- Making simulations efficient
- Bayesian posterior estimation

Today:

- Using simulation f inference
- Understanding est
- Real-world applica
- Hands-on experim

🎯Today's Big Question

When we estimate something from data,

how do we know if our method is any go

MC Inference: The Core Idea

Traditional Statistics

- Derive formulas analytically
- Make distributional assumptions
- Hope assumptions hold!
- Limited to simple cases

Example: "Assume normality, then the sample mean has variance σ^2/n"

Monte Carlo Ap

- Simulate the data-
- Compute estimato
- Observe actual be
- Works for complex

Example: "Let's generat and see what actually

💡 Key Insight

MC lets us study estimator properties empirically **when theo**

Part 1: Estimating Expected V

Motivating Example: Difference of Normals

Setup: $X_1, X_2 \sim N(0, 1)$ independently

Question: What is $E[|X_1 - X_2|]$?

🤔Think-Pair-Share (2 minutes)

1. **Think:** Can you compute this analytically? What distribution does $|X$
2. **Pair:** Discuss with your neighbor
3. **Share:** Let's hear some thoughts!

The MC Answer: Just Simulate It!

```
m ← 10000
X1 ← rnorm(m)
X2 ← rnorm(m)
theta_hat ← mean(abs(X1 - X2))
theta_hat  # Our estimate
```

Code: 🖥️

```r
set.seed(123)
m ← 10000

# Generate samples
X1 ← rnorm(m)
X2 ← rnorm(m)

# Compute estimator
theta_hat ← mean(abs(X1 - X2))

# Standard error
se_theta ← sd(abs(X1 - X2)) / sqrt(m)

# Results
cat("Estimate:", round(theta_hat, 4), "\n")
```

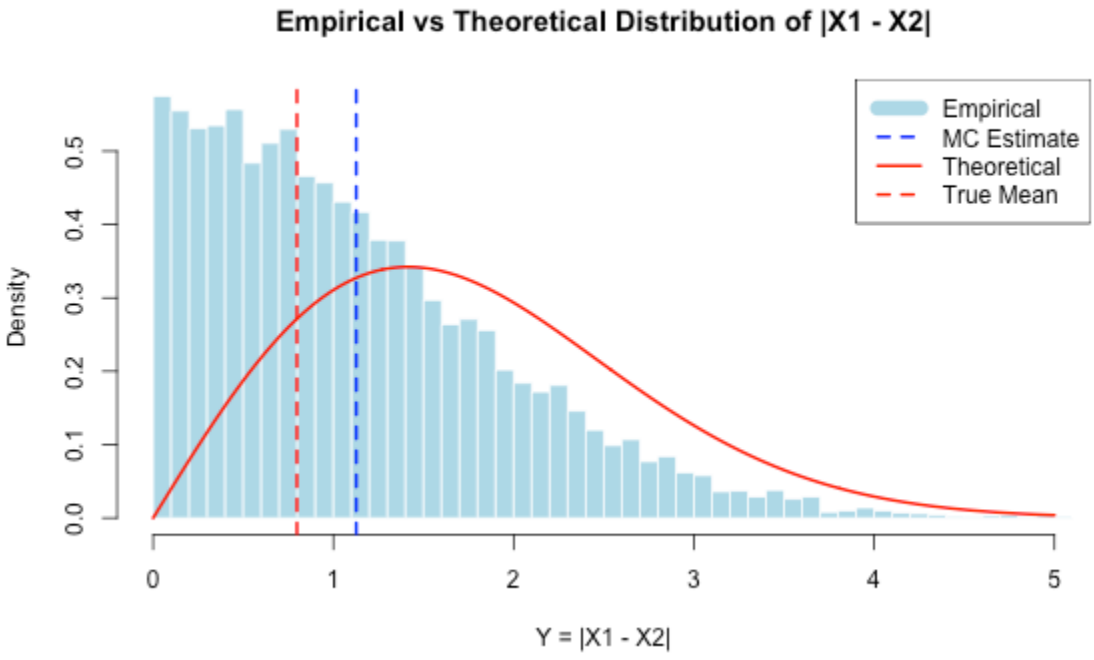
```
## Estimate: 1.1268
```

```r
cat("SE:", round(se_theta, 5), "\n")
```

```
## SE: 0.00848
```

```r
cat("95% CI: [", round(theta_hat - 1.96*se_theta, 4), ",",
    round(theta_hat + 1.96*se_theta, 4), "]\n")
```

Visualizing the Distribution



Estimating Probabilities

Question: What is $P[|X_1 - X_2| > 1]$?

Key Insight: Any probability is an expected value!

$$P[Y > 1] = E[1_{Y>1}]$$

where $1_{Y>1}$ is the indicator function.

MC Algorithm

```
m ← 100000
Y ← abs(rnorm(m) - rnorm(m))

# Estimate probability
p_hat ← mean(Y > 1)
se_p ← sqrt(p_hat * (1 - p_hat) / m)

cat("P[Y > 1] ≈", round(p_hat, 4), "±", round(se_p, 4))
```

```
## P[Y > 1] ≈ 0.4818 ± 0.0016
```

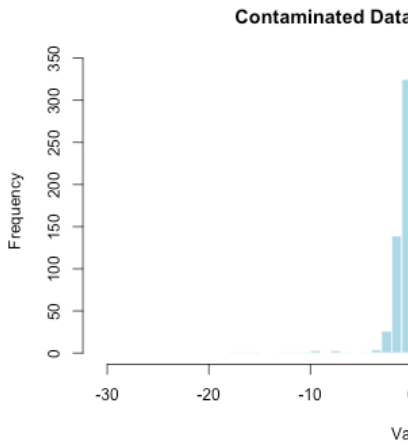
Quick Poll

Part 2: Comparing Estimators w

The Contaminated Data Problem

Scenario:

- Most data from $N(0, 1)$
- Some contaminated from $N(0, 100)$
- Want to estimate the mean (should be 0)



Contamination model:

$$pN(0, 1) + (1 - p)N(0, 100)$$

Question: Should we use $\bar{X}$ or something more robust?

Trimmed Mean to the Rescue

Idea: Remove extreme values before computing mean

$$\bar{X}_{[-k]} = \frac{1}{n-2k} \sum_{i=k+1}^{n-k} X_{(i)}$$

where $X_{(1)} \leq X_{(2)} \leq \cdots \leq X_{(n)}$ are ordered observations.

🎯Your Turn! (Think)

Before we simulate, make a prediction:

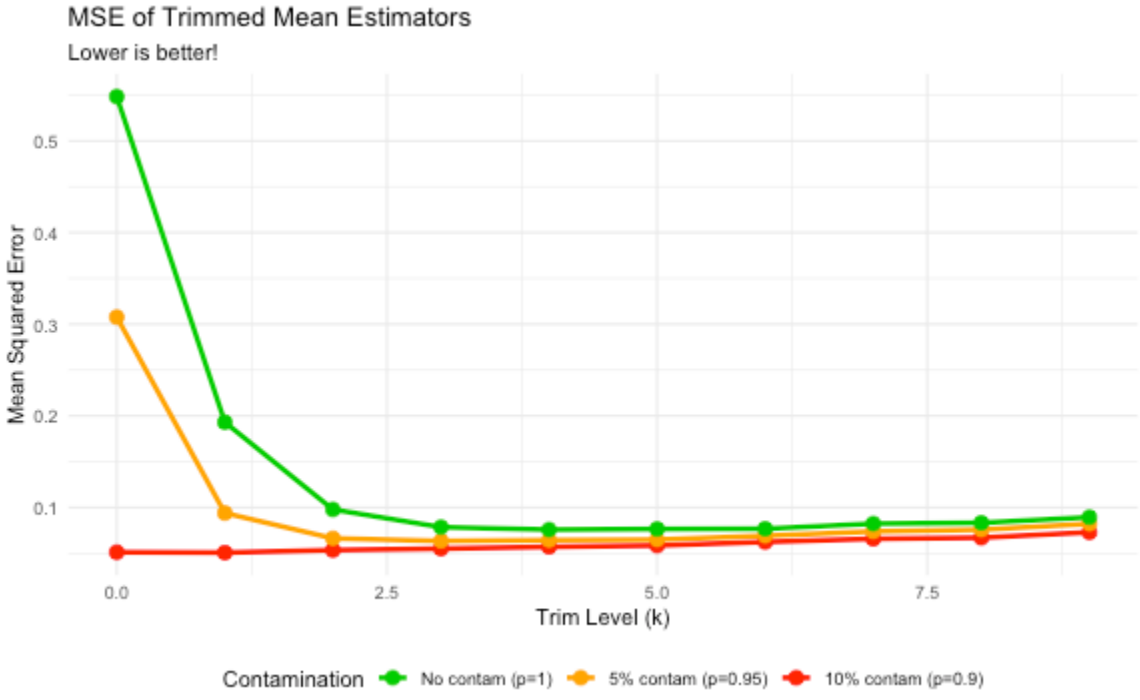
1. When there's **no contamination** (p=1), which is better: $\bar{X}$ or $\bar{X}_{[-k]}$?
2. When there's **5% contamination** (p=0.95), which is better?
3. What happens as we increase the trim level k?

Now let's find out with simulation!

Code: Comparing Estimators

```r
# Function to compute MSE of trimmed mean
trimmed_mse ← function(n, m, k, p) {
  tmean ← numeric(m)

  for(i in 1:m) {
    # Generate contaminated sample
    sigma ← sample(c(1, 10), size = n, replace = TRUE,
                   prob = c(p, 1-p))
    x ← rnorm(n, 0, sigma)
    x_sorted ← sort(x)

    # Compute trimmed mean
    if(k == 0) {
      tmean[i] ← mean(x_sorted)
    } else {
      tmean[i] ← mean(x_sorted[(k+1):(n-k)])
    }
  }

  # MSE (true mean is 0)
  mse ← mean(tmean^2)
  se_mse ← sd(tmean^2) / sqrt(m)

  return(c(mse = mse, se = se_mse))
}
```

Results: MSE for Different Scena



What Did We Learn? 💡

Table: MSE by Trim Level and Contamination Rate

k	0%	5%	10%
0	0.0512	0.3077	0.5484
1	0.0508	0.0939	0.1931
2	0.0535	0.0663	0.0978
3	0.0551	0.0635	0.0785
4	0.0570	0.0644	0.0758
5	0.0585	0.0649	0.0765
6	0.0624	0.0691	0.0768
7	0.0658	0.0738	0.0824
8	0.0670	0.0758	0.0833
9	0.0729	0.0822	0.0894

Key Insights:

Part 3: Power Analysis for Expe

The Deliveroo Problem

Business Context:

- Testing new app feature
- Measure: order completion rate
- Question: **How many users do we need to detect a 3% improvement?**

Classical Approach:

- Use formula for proportion test
- Assumes normality (CLT)
- Ignores real data quirks

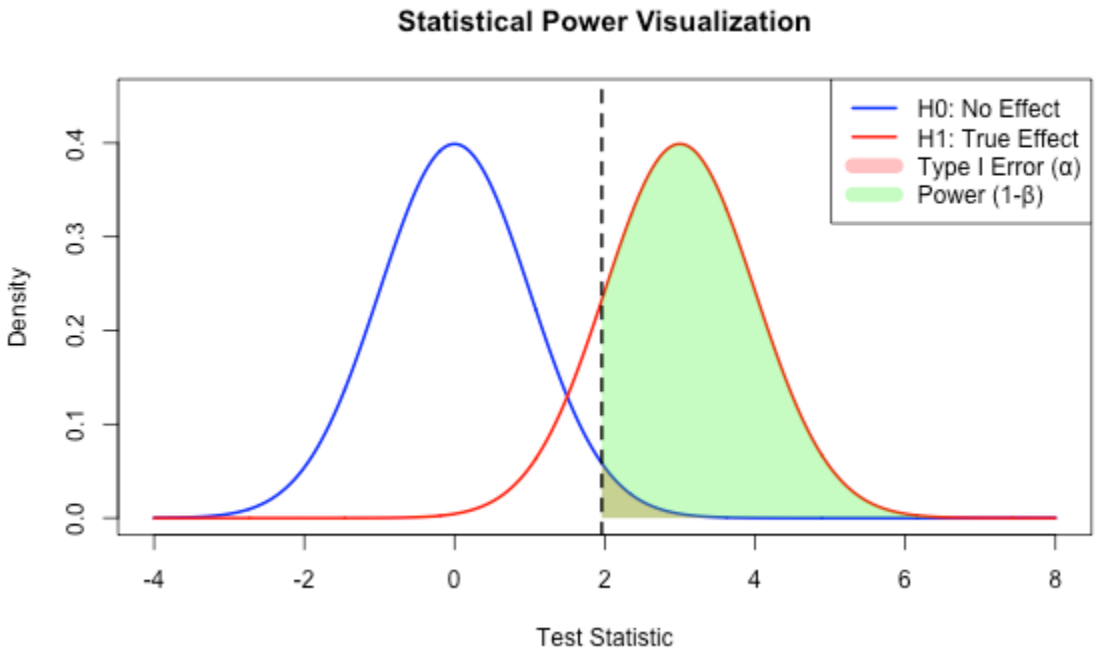
MC Approach:

- Use actual historic
- Simulate experime
- Account for real di
- More accurate sam

🎯This is REAL

Companies like Deliver
use this daily.

What is Statistical Power?



Power = Probability of detecting a true effect

MC Power Analysis: The Algorith

Steps:

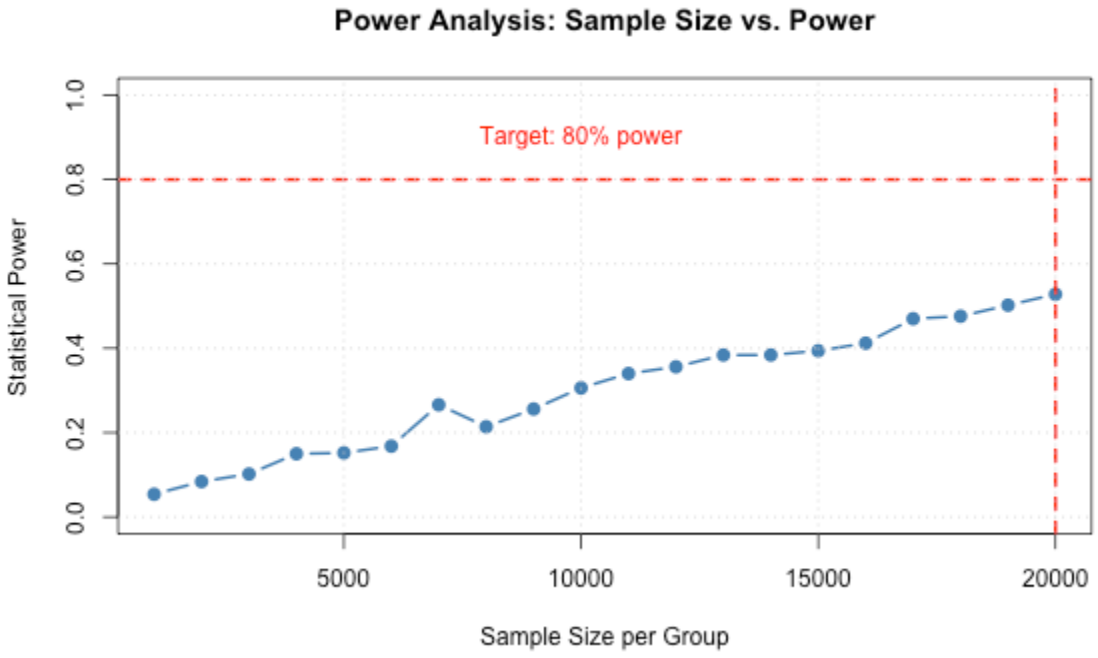
1. **Collect historical data** on your metric
2. **Choose** your experiment parameters:
    - Sample size to test (N)
    - Effect size you want to detect (δ)
    - Significance level (α = 0.05)
3. **For each sample size** (repeat many times):
    - Generate control group from historical data
    - Generate treatment group with effect applied
    - Run statistical test
    - Record if test was significant
4. **Power = fraction of significant results**

Live Example: Testing Conversio

```r
# Historical data: 31.5% conversion rate
base_rate ← 0.315
target_effect ← 1.03  # Want to detect 3% lift
alpha ← 0.05
sims ← 1000

# Function to estimate power for given sample size
estimate_power ← function(n_per_group, base_rate, effect, sims) {
  significant ← numeric(sims)

  for(i in 1:sims) {
    # Generate data
    control ← rbinom(n_per_group, 1, base_rate)
    treatment ← rbinom(n_per_group, 1, base_rate * effect)

    # Run test
    test ← prop.test(c(sum(control), sum(treatment)),
                     c(n_per_group, n_per_group))

    significant[i] ← test$p.value < alpha
  }

  return(mean(significant))
}
```

Building the Power Curve

```
# Test different sample sizes
sample_sizes ← seq(1000, 20000, 1000)
power_estimates ← numeric(length(sample_sizes))

set.seed(42)
for(i in seq_along(sample_sizes)) {
  power_estimates[i] ← estimate_power(sample_sizes[i],
                                      base_rate, target_effect, 5
}
```



Result: Need ~2\times 10^{4} users per group!

🛠️Time to Get Your Hands

Activity: Trimmed Mean Analysis

Break into groups of 2-3

Activity Overview

You'll investigate three questions:

1️⃣ Median vs Trimmed Mean

How does the sample median perform compared to different trim levels

2️⃣ Increased Contamination

What happens when contamination rate increases to 40%?

3️⃣ Your Own Question!

Design and run your own MC study:

- Different distributions?
- Different estimators?
- Different contamination patterns?

Time: 30 minutes
Deliverable: Brief presentation (3 minutes) of findings

Activity Tips & Resources

📋 Worksheet provided with:

- Starter code
- Guiding questions
- Space for your discoveries

💡 Remember to:

1. Start with intuition (what do you expect?)
2. Design your MC study carefully
3. Interpret results, don't just report numbers
4. Compare to theory when possible

🆘 Need help?

- Check the code examples from slides
- Consult with neighboring groups
- Ask instructor!

Wrap-Up & Takeaways

What You Can Do with MC Infere

✅Things we covered:

1. Estimate properties of estimators (bias, variance, MSE)
2. Compare different estimation methods
3. Compute probabilities for complex distributions
4. Design proper experiments (power analysis)

🚀Beyond this class:

- Bootstrap and resampling methods
- Bayesian credible intervals
- Sequential testing
- Adaptive designs

💼Real-world impact:

This is literally how companies make million-dollar decisions!

Key Insights

Why MC Inference Matters

1. **Flexibility:** Works when formulas don't exist
2. **Realism:** Use actual data distributions
3. **Transparency:** See what's happening
4. **Scalability:** Same approach for simple & complex problems

Best Practices

1. ✓ Always check a f intuition
2. ✓ Report standard
3. ✓ Visualize results
4. ✓ Compare to theo
5. ✓ Document your parameters

For Next Time

📚Recommended Reading:

- Deliveroo article (linked in worksheet)

Questions?

Extra: Keep exploring! 

```r
# Example: Build your own MC study
my_mc_study ← function(estimator, data_generator,
                       true_value, n_sims = 10000) {
  estimates ← replicate(n_sims, {
    data ← data_generator()
    estimator(data)
  })

  list(
    bias = mean(estimates) - true_value,
    variance = var(estimates),
    mse = mean((estimates - true_value)^2)
  )
}
```