Code

Monte Carlo Inference Activity

Exploring Robust Estimators through Simulation

PSTAT 194CS

2026-02-08

Introduction

In this activity, you'll conduct your own Monte Carlo studies to understand how different estimators perform under contamination. You'll work through three exercises, building from guided exploration to independent investigation.

Learning Objectives

By the end of this activity, you will be able to:

1. Design and implement Monte Carlo simulation studies
2. Compare estimator performance using MSE
3. Interpret simulation results in context
4. Formulate and test statistical hypotheses empirically

The Contamination Model

Throughout these exercises, we'll work with the contamination model:

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

This represents data where:

- A proportion p of observations come from $N(0, 1)$ (good data)
- A proportion $(1 - p)$ come from $N(0, 100)$ (contaminated/outliers)
- True mean we're trying to estimate: **0**

Exercise 1: Trimmed Mean Estimator

Part A: Understanding the Code

First, let's review the function for computing MSE of the trimmed mean:

Hide

```
# Function to compute MSE of k-level trimmed mean
trimmed_mse <- function(n, m, k, p) {
  # n = sample size per iteration
  # m = number of MC iterations
  # k = trim level (removes k smallest and k largest)
  # p = proportion of good data (1-p is contamination rat
      e)

  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)  # Regular mean
    } else {
      tmean[i] <- mean(x_sorted[(k+1):(n-k)])  # Trimmed
        mean
    }
  }

  # MSE calculation (true mean = 0)
  mse <- mean(tmean^2)
  se_mse <- sd(tmean^2) / sqrt(m)

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

Understanding Check (discuss with your group):

1. **Why do we use `tmean^2` to compute MSE?**

   Hint: What is MSE when the true parameter is 0?

2. **What does k=0 represent?**

▶ Click for answer

3. **If n=20 and k=5, how many observations are we using?**

▶ Click for answer

Part B: Run the Basic Study

Let's replicate the analysis from class:

Hide

```
set.seed(194)  # For reproducibility

# Study parameters
n <- 20        # Sample size
m <- 10000     # MC iterations
K <- n/2 - 1   # Maximum trim level

# Storage for results
results <- expand.grid(k = 0:9, p = c(1, 0.95, 0.9))
results$mse <- NA
results$se <- NA

# Run simulations
for(i in 1:nrow(results)) {
  res <- trimmed_mse(n, m, results$k[i], results$p[i])
  results$mse[i] <- res[1]
  results$se[i] <- res[2]
}

# View results
results %>%
  mutate(contamination = paste0((1-p)*100, "%"),
         mse = round(mse, 4),
         se = round(se, 5)) %>%
  select(k, contamination, mse, se) %>%
  pivot_wider(names_from = contamination,
              values_from = c(mse, se)) %>%
  kable(caption = "MSE (and SE) by Trim Level and Contami
        nation") %>%
  kable_styling(bootstrap_options = c("striped", "hove
        r"))
```

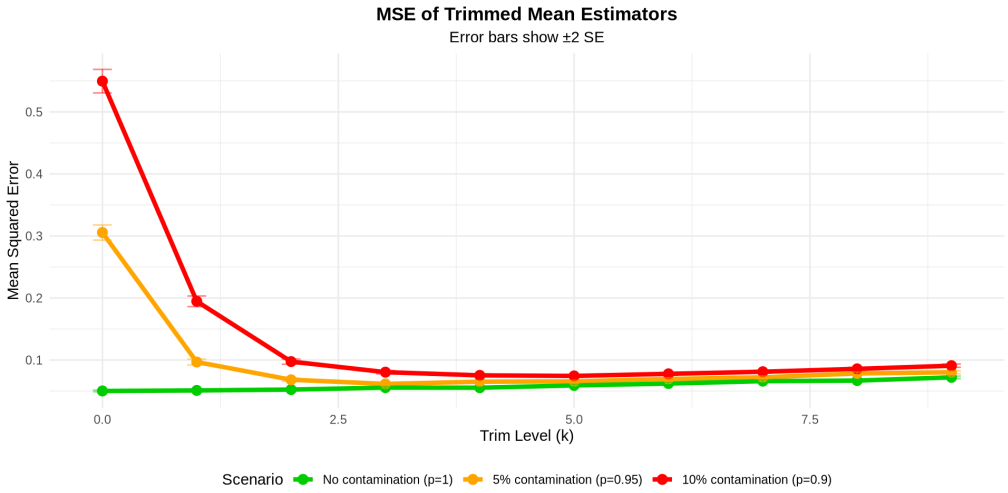
MSE (and SE) by Trim Level and Contamination

k	mse_0%	mse_5%	mse_10%	se_0%	se_5%	se_10%
0	0.0501	0.3055	0.5496	0.00069	0.00611	0.00950
1	0.0510	0.0968	0.1946	0.00073	0.00238	0.00429
2	0.0524	0.0682	0.0975	0.00074	0.00132	0.00208
3	0.0555	0.0615	0.0804	0.00079	0.00091	0.00137
4	0.0553	0.0650	0.0752	0.00078	0.00093	0.00110
5	0.0590	0.0656	0.0743	0.00085	0.00092	0.00108
6	0.0620	0.0694	0.0777	0.00090	0.00101	0.00110
7	0.0659	0.0721	0.0811	0.00096	0.00103	0.00116
8	0.0668	0.0784	0.0858	0.00096	0.00113	0.00126
9	0.0720	0.0802	0.0908	0.00102	0.00114	0.00132

Part C: Visualize Results

Hide

```
ggplot(results, aes(x = k, y = mse, color = factor(p, lev
        els = c(1, 0.95, 0.9)), group = p)) +
  geom_line(size = 1.5) +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = mse - 2*se, ymax = mse + 2*s
        e),
                width = 0.2, alpha = 0.5) +
  scale_color_manual(values = c("green3", "orange", "re
        d"),
                     labels = c("No contamination (p=1)",
                                "5% contamination (p=0.9
        5)",
                                "10% contamination (p=0.
        9)")) +
  labs(title = "MSE of Trimmed Mean Estimators",
       subtitle = "Error bars show ±2 SE",
       x = "Trim Level (k)",
       y = "Mean Squared Error",
       color = "Scenario") +
  theme_minimal() +
  theme(legend.position = "bottom",
        plot.title = element_text(hjust = 0.5, face = "bo
        ld"),
        plot.subtitle = element_text(hjust = 0.5))
```



Part D: Interpret the Results

Questions for Discussion:

1. **What happens to MSE as we increase k when there's NO contamination (p=1)?**

```
MSE slightly increases as we increase increase k and
there is no contaminatrion.
```

2. **What trim level (k) gives the best (lowest) MSE for 5% contamination?**

```
k = 3 gives us the lowest MSE for 5% contamination.
```

3. **Why does this optimal trim level make sense?**

   Hint: Think about how many contaminated observations you expect in a sample of 20.

```
  We would expect one contaminated sample, so I do n
ot know why this makes sense.
```

Exercise 2: Sample Median vs Trimmed Mean

Part A: Formulate Your Hypothesis

Before running any code, think about how the sample median might perform.

Intuition Building:

1. **The median is equivalent to what trim level?**

   Hint: For n=20, the median uses which observations?

```
The. median would be be equivalent to the trimmed me
an when k = 9.
```

2. **Prediction: Will the median be more or less robust than low-trim estimators?**

```
  Median has higher variance so the estimator will b
e less robust than low-trim estimators.
```

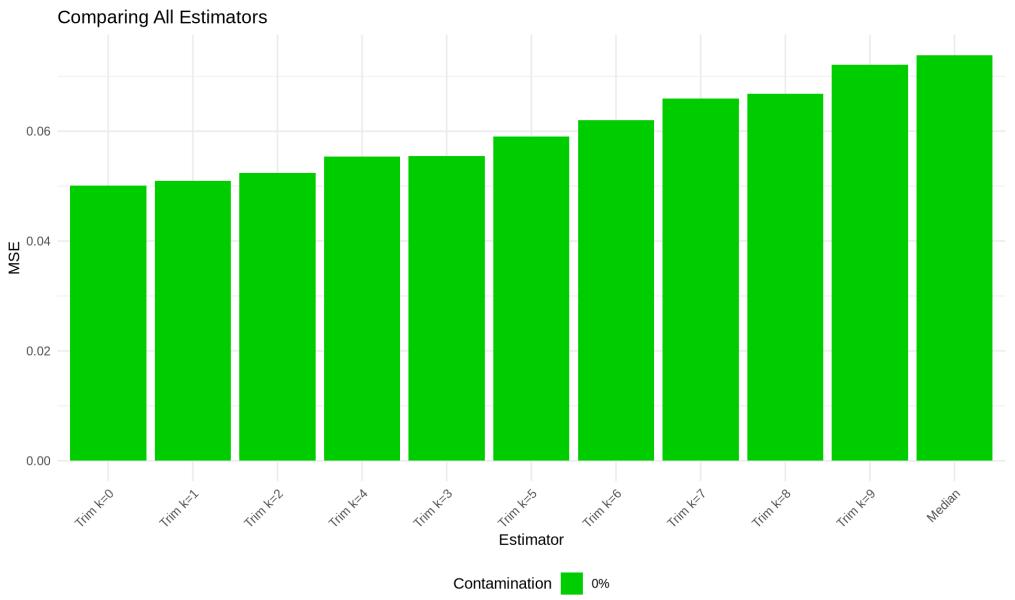
Part B: Implement Median Comparison

Now modify the code to include the median:

Hide

```
# Function to compute MSE for median
median_mse <- function(n, m, p) {
  med <- numeric(m)

  for(i in 1:m) {
    sigma <- sample(c(1, 10), size = n, replace = TRUE,
                    prob = c(p, 1-p))
    x <- rnorm(n, 0, sigma)

    med[i] <- median(x)
  }

  mse <- mean(med^2)
  se_mse <- sd(med^2) / sqrt(m)

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

# Add median results to our previous results
set.seed(194)
median_results <- data.frame(
  k = rep("Median", 3),
  p = c(1, 0.95, 0.9)
)

for(i in 1:3) {
  res <- median_mse(20, 10000, median_results$p[i])
  median_results$mse[i] <- res[1]
  median_results$se[i] <- res[2]
}

# Combine with previous results
results_all <- results %>%
  mutate(estimator = paste0("Trim k=", k),
         k = as.character(k)) %>%
  bind_rows(median_results %>% mutate(estimator = "Media
        n")) %>%
  mutate(contamination = (1-p)*100)
```

```
# Plot comparison
results_all %>%
  filter(contamination %in% c(0, 5, 10)) %>%
  ggplot(aes(x = reorder(estimator, mse), y = mse, fill =
        factor(contamination))) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_fill_manual(values = c("green3", "orange", "re
        d"),
                    labels = c("0%", "5%", "10%")) +
  labs(title = "Comparing All Estimators",
       x = "Estimator", y = "MSE", fill = "Contaminatio
        n") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust =
        1),
        legend.position = "bottom")
```



Part C: Analyze and Discuss

Create a table comparing the median to other estimators:

Hide

```
library(dplyr)
comparison_table <- results_all %>%
  filter(contamination == "5") %>%
  select(estimator, mse, se) %>%
  arrange(mse) %>%
  mutate(mse = round(mse, 4),
         se = round(se, 5),
         rank = row_number())

comparison_table %>%
  kable(caption = "Estimator Rankings for 5% Contaminatio
        n") %>%
  kable_styling(bootstrap_options = c("striped", "hove
        r")) %>%
  row_spec(which(comparison_table$estimator == "Median"),
           bold = TRUE, background = "yellow")
```

Estimator Rankings for 5% Contamination

estimator	mse	se	rank
Trim k=3	0.0615	0.00091	1
Trim k=4	0.0650	0.00093	2
Trim k=5	0.0656	0.00092	3
Trim k=2	0.0682	0.00132	4
Trim k=6	0.0694	0.00101	5
Trim k=7	0.0721	0.00103	6
Trim k=8	0.0784	0.00113	7
Trim k=9	0.0802	0.00114	8
Median	**0.0803**	**0.00113**	**9**
Trim k=1	0.0968	0.00238	10
Trim k=0	0.3055	0.00611	11

Discussion Questions:

1. **Where does the median rank?**

```
 Median is ranked 9th.
```

2. **Did this match your prediction from Part A?**

```
 Yes this did match my prediction. I predicted that
because of the high variance of median that it would
have a high MSE.
```

3. **The median is a special case of which trimmed mean?**

```
   It is the spacial case of the trimmed mean where
$k = n/2 - 1$.
```

4. **Would the median work well for ANY distribution? Why or why not?**

   Think about: What if the true distribution is very skewed?

```
The median would not work well if the distributon wa
s not centered around 0 like exponential for exampl
e. This would not work well becaause there is lots o
f data to the right  and no data do the left, but we
trim from both sides.
```

Part D: Theoretical Verification

For the uncontaminated case, we can compare to theory:

Hide

```
# Theoretical values for N(0,1) with n=20
n <- 20

# Mean
var_mean <- 1/n  # σ²/n
mse_mean_theory <- var_mean

# Median (approximate for normal)
var_median_approx <- (pi/2) * (1/n)  # ≈ 1.57/n
mse_median_theory <- var_median_approx

cat("Theoretical MSE (no contamination, n=20):\n")
```

```
## Theoretical MSE (no contamination, n=20):
```

Hide

```
cat("Mean:  ", round(mse_mean_theory, 4), "\n")
```

```
## Mean:   0.05
```

Hide

```
cat("Median:", round(mse_median_theory, 4), "\n\n")
```

```
## Median: 0.0785
```

Hide

```
cat("Our MC Estimates:\n")
```

```
## Our MC Estimates:
```

Hide

```
cat("Mean (k=0):", round(results_all$mse[results_all$esti
        mator == "Trim k=0" &
                                          results_all$con
        tamination == 0], 4), "\n")
```

```
## Mean (k=0): 0.0501
```

Hide

```
cat("Median:    ", round(results_all$mse[results_all$esti
        mator == "Median" &
                                          results_all$con
        tamination == 0], 4), "\n")
```

```
## Median:     0.0738
```

Do they match? If not, why might there be small differences? There might be small differences because when trimming the mean we might be trimming actual uncontaminated samples/ —

Exercise 3: Effect of Increased Contamination

Part A: Make Predictions

You're now going to study what happens with **40% contamination** (p=0.6).

🎯 Before coding, predict:

1. **Will trimming help more or less than with 5% contamination?**

   Your hypothesis:

   ```
   Trimming will help more becasue there is more contam
   inated data
   ```

2. **What trim level do you think will be optimal?**

   Your reasoning:

   ```
   I think 50% trim weill be optimal becaseu when we we
   re working with 5% contamination we needed extra to
   effectively reduce mean square error.
   ```

3. **How will the regular mean (k=0) perform?**

   Your prediction:

   ```
   It will perform very poorly.
   ```

Part B: Run the Study

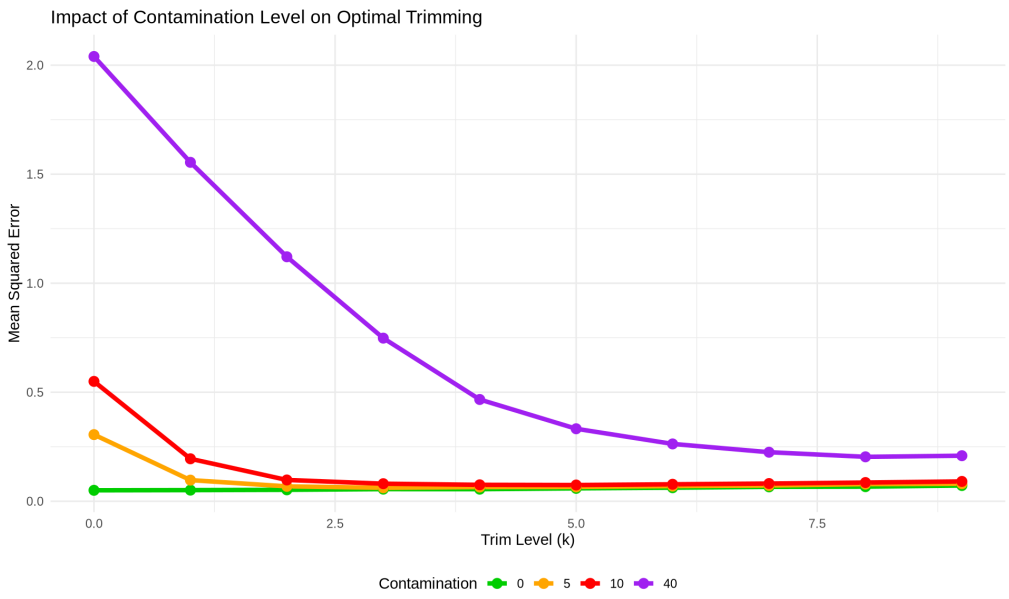
Hide

```r
set.seed(194)

# YOUR CODE HERE
# Modify the parameters to study p=0.6 (40% contaminatio
      n)
# Compare to previous results

# Hint: You can copy and modify the code from Exercise 1

# Study parameters
n <- 20
m <- 10000

# Run for p=0.6
high_contam_results <- expand.grid(k = 0:9, p = 0.6)
high_contam_results$mse <- NA
high_contam_results$se <- NA

for(i in 1:nrow(high_contam_results)) {
  res <- trimmed_mse(n, m, high_contam_results$k[i], high
        _contam_results$p[i])
  high_contam_results$mse[i] <- res[1]
  high_contam_results$se[i] <- res[2]
}

# Add median
median_40 <- median_mse(20, 10000, 0.6)
```

Part C: Visualization and Comparison

Hide

```
# Combine all contamination levels
all_levels <- results %>%
  mutate(k = as.character(k)) %>%
  bind_rows(high_contam_results %>% mutate(k = as.charact
        er(k))) %>%
  mutate(contamination_pct = (1-p)*100)

# Plot all levels
ggplot(all_levels, aes(x = as.numeric(k), y = mse,
                       color = factor(contamination_pct),
                       group = contamination_pct)) +
  geom_line(size = 1.5) +
  geom_point(size = 3) +
  scale_color_manual(values = c("green3", "orange", "re
        d", "purple"),
                     name = "Contamination") +
  labs(title = "Impact of Contamination Level on Optimal
        Trimming",
       x = "Trim Level (k)",
       y = "Mean Squared Error") +
  theme_minimal() +
  theme(legend.position = "bottom")
```



Part D: Analysis Questions

1. **What trim level is optimal for 40% contamination?**

Hide

```
optimal_k <- high_contam_results$k[which.min(high_cont
     am_results$mse)]
cat("Optimal k for 40% contamination:", optimal_k, "\
     n")
```

```
## Optimal k for 40% contamination: 8
```

Hide

```
cat("MSE at optimal k:",
    round(min(high_contam_results$mse), 4), "\n")
```

```
## MSE at optimal k: 0.2036
```

Your interpretation:

```
If you trim 8 from each end or 80% percent of the data yo
ur mean square error is the lowers.
```

2. **How much worse is the regular mean (k=0) at 40% vs 5% contamination?**

YOUR CODE: Compare MSE of k=0 at different contamination levels

Hide

```
results$mse[1]
```

```
## [1] 0.05012024
```

Hide

```
high_contam_results$mse[1]
```

```
## [1] 2.039864
```

Hide

Hide

```
dif <- high_contam_results$mse[1] - results$mse[1]
dif
```

```
## [1] 1.989744
```

3. **Create a summary table:**

Hide

```
# Table showing optimal trim level for each contaminat
    ion rate
summary_optimal <- all_levels %>%
  group_by(contamination_pct) %>%
  summarize(
    optimal_k = k[which.min(mse)],
    min_mse = min(mse),
    mean_k0_mse = mse[k == "0"]
  ) %>%
  mutate(benefit_of_trimming = mean_k0_mse / min_mse)

summary_optimal %>%
  mutate(across(where(is.numeric), ~round(., 3))) %>%
  kable(caption = "Optimal Trimming Strategy by Contam
    ination Level") %>%
  kable_styling(bootstrap_options = c("striped", "hove
    r"))
```

Optimal Trimming Strategy by Contamination Level

contamination_pct	optimal_k	min_mse	mean_k0_mse	benefit_of_trimming
0	0	0.050	0.050	1.000
5	3	0.061	0.306	4.969
10	5	0.074	0.550	7.393
40	8	0.204	2.040	10.017

4. **Did your predictions from Part A match the results?**

   Reflection:

```
No we had to trim a lot more than I expected.
```

Exercise 4: Design Your Own Study! 🚀

Now it's time to explore something that interests you!

Part A: Choose Your Research Question

Here are some ideas (or create your own):

Option 1: Different Distributions

What if the uncontaminated data isn't normal? Try: - Exponential distribution - Uniform distribution - t-distribution with low df

Option 2: Different Estimators

Compare other robust estimators: - Winsorized mean (replace extremes instead of removing) - Interquartile mean (trim to 25th-75th percentiles) - Huber M-estimator

Option 3: Different Contamination Patterns

What if contamination isn't symmetric? - One-sided contamination (only large values) - Multiple contamination sources - Increasing contamination with sample size

Option 4: Practical Application

Apply to real data: - Stock returns with crashes - Sensor data with malfunctions - Survey data with trolls

Your chosen question:

```
I am going to try this with exponential distribution.
```

Part B: Design Your Study

Study Design Plan:

1. **What are you testing?**

   I am testing how effectively trimming works for one sided distributions specifically exponential.

2. **What parameters will you vary?**

   I will vary the distribution.

3. **How will you measure performance?**

   I will measure the reduction in mean square error at certain levels of contamination.

4. **What do you expect to find?**

   I expect to find trimming is less effective for one sided distributions.

Part C: Implement Your Study

Hide

```
# YOUR CODE HERE
# Design and implement your Monte Carlo study

set.seed(194)  # Keep for reproducibility

# Example template:
diff_funcs <- function(n, m, k , p, dist_func) {
  # Define your parameters

  # n = sample size per iteration
  # m = number of MC iterations
  # k = trim level (removes k smallest and k largest)
  # p = proportion of good data (1-p is contamination rat
      e)
  # dist_pdf = the distribution that we are trying
```

```
  tmean <- numeric(m)

  for(i in 1:m) {
    # Generate contaminated sample
    if (identical(dist_func, rnorm)){
      param <- sample(c(1, 10), size = n, replace = TRUE,
                      prob = c(p, 1-p))
      x <- dist_func(n, 0, param)
    }
    if (identical(dist_func, rexp)){
      param <- sample(c(1, 10), size = n, replace = TRUE,
                      prob = c(p, 1-p))
      x <- dist_func(n, param)
    }

    x_sorted <- sort(x)

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

  # MSE calculation (true mean = 0)
  true_mean <- if (identical(dist_func, rexp)) {
    0.1 + 0.9*p   # because means are 1 (rate=1) and 0.1
        (rate=10)
  } else {
   0
  }

  mse <- mean((tmean - true_mean)^2)
  se_mse <- sd((tmean - true_mean)^2) / sqrt(m)

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

Hide

```
# Run your study
run_tests <- function(n = 20, m = 10000,  dist_func){
  set.seed(194)  # For reproducibility

K = n/2 - 1

# Storage for results
results <- expand.grid(k = 0:9, p = c(1, 0.95, 0.9))
results$mse <- NA
results$se <- NA

# Run simulations
for(i in 1:nrow(results)) {
  res <- diff_funcs(n, m, results$k[i], results$p[i], dis
        t_func)
  results$mse[i] <- res[1]
  results$se[i] <- res[2]
}

# View results
tbl <- results %>%
  mutate(contamination = paste0((1-p)*100, "%"),
         mse = round(mse, 4),
         se = round(se, 5)) %>%
  select(k, contamination, mse, se) %>%
  pivot_wider(names_from = contamination,
              values_from = c(mse, se)) %>%
  kable(caption = "MSE (and SE) by Trim Level and Contami
        nation") %>%
  kable_styling(bootstrap_options = c("striped", "hove
        r"))

print(tbl)
return(results)
}
```

Hide

```
norm_test <- run_tests(dist_func = rnorm)
```

```
## <table class="table table-striped table-hover" style="
margin-left: auto; margin-right: auto;">
## <caption>MSE (and SE) by Trim Level and Contamination
</caption>
```

```
##  <thead>
##   <tr>
##    <th style="text-align:right;"> k </th>
##    <th style="text-align:right;"> mse_0% </th>
##    <th style="text-align:right;"> mse_5% </th>
##    <th style="text-align:right;"> mse_10% </th>
##    <th style="text-align:right;"> se_0% </th>
##    <th style="text-align:right;"> se_5% </th>
##    <th style="text-align:right;"> se_10% </th>
##   </tr>
##  </thead>
## <tbody>
##   <tr>
##    <td style="text-align:right;"> 0 </td>
##    <td style="text-align:right;"> 0.0501 </td>
##    <td style="text-align:right;"> 0.3055 </td>
##    <td style="text-align:right;"> 0.5496 </td>
##    <td style="text-align:right;"> 0.00069 </td>
##    <td style="text-align:right;"> 0.00611 </td>
##    <td style="text-align:right;"> 0.00950 </td>
##   </tr>
##   <tr>
##    <td style="text-align:right;"> 1 </td>
##    <td style="text-align:right;"> 0.0510 </td>
##    <td style="text-align:right;"> 0.0968 </td>
##    <td style="text-align:right;"> 0.1946 </td>
##    <td style="text-align:right;"> 0.00073 </td>
##    <td style="text-align:right;"> 0.00238 </td>
##    <td style="text-align:right;"> 0.00429 </td>
##   </tr>
##   <tr>
##    <td style="text-align:right;"> 2 </td>
##    <td style="text-align:right;"> 0.0524 </td>
##    <td style="text-align:right;"> 0.0682 </td>
##    <td style="text-align:right;"> 0.0975 </td>
##    <td style="text-align:right;"> 0.00074 </td>
##    <td style="text-align:right;"> 0.00132 </td>
##    <td style="text-align:right;"> 0.00208 </td>
##   </tr>
##   <tr>
##    <td style="text-align:right;"> 3 </td>
##    <td style="text-align:right;"> 0.0555 </td>
##    <td style="text-align:right;"> 0.0615 </td>
##    <td style="text-align:right;"> 0.0804 </td>
##    <td style="text-align:right;"> 0.00079 </td>
```

```
##    <td style="text-align:right;"> 0.00091 </td>
##    <td style="text-align:right;"> 0.00137 </td>
##   </tr>
##   <tr>
##    <td style="text-align:right;"> 4 </td>
##    <td style="text-align:right;"> 0.0553 </td>
##    <td style="text-align:right;"> 0.0650 </td>
##    <td style="text-align:right;"> 0.0752 </td>
##    <td style="text-align:right;"> 0.00078 </td>
##    <td style="text-align:right;"> 0.00093 </td>
##    <td style="text-align:right;"> 0.00110 </td>
##   </tr>
##   <tr>
##    <td style="text-align:right;"> 5 </td>
##    <td style="text-align:right;"> 0.0590 </td>
##    <td style="text-align:right;"> 0.0656 </td>
##    <td style="text-align:right;"> 0.0743 </td>
##    <td style="text-align:right;"> 0.00085 </td>
##    <td style="text-align:right;"> 0.00092 </td>
##    <td style="text-align:right;"> 0.00108 </td>
##   </tr>
##   <tr>
##    <td style="text-align:right;"> 6 </td>
##    <td style="text-align:right;"> 0.0620 </td>
##    <td style="text-align:right;"> 0.0694 </td>
##    <td style="text-align:right;"> 0.0777 </td>
##    <td style="text-align:right;"> 0.00090 </td>
##    <td style="text-align:right;"> 0.00101 </td>
##    <td style="text-align:right;"> 0.00110 </td>
##   </tr>
##   <tr>
##    <td style="text-align:right;"> 7 </td>
##    <td style="text-align:right;"> 0.0659 </td>
##    <td style="text-align:right;"> 0.0721 </td>
##    <td style="text-align:right;"> 0.0811 </td>
##    <td style="text-align:right;"> 0.00096 </td>
##    <td style="text-align:right;"> 0.00103 </td>
##    <td style="text-align:right;"> 0.00116 </td>
##   </tr>
##   <tr>
##    <td style="text-align:right;"> 8 </td>
##    <td style="text-align:right;"> 0.0668 </td>
##    <td style="text-align:right;"> 0.0784 </td>
##    <td style="text-align:right;"> 0.0858 </td>
##    <td style="text-align:right;"> 0.00096 </td>
```

```
##    <td style="text-align:right;"> 0.00113 </td>
##    <td style="text-align:right;"> 0.00126 </td>
##   </tr>
##   <tr>
##    <td style="text-align:right;"> 9 </td>
##    <td style="text-align:right;"> 0.0720 </td>
##    <td style="text-align:right;"> 0.0802 </td>
##    <td style="text-align:right;"> 0.0908 </td>
##    <td style="text-align:right;"> 0.00102 </td>
##    <td style="text-align:right;"> 0.00114 </td>
##    <td style="text-align:right;"> 0.00132 </td>
##   </tr>
## </tbody>
## </table>
```

Hide

```
exp_test <- run_tests(dist_func = rexp)
```

```
## <table class="table table-striped table-hover" style="
margin-left: auto; margin-right: auto;">
## <caption>MSE (and SE) by Trim Level and Contamination
</caption>
##  <thead>
##   <tr>
##    <th style="text-align:right;"> k </th>
##    <th style="text-align:right;"> mse_0% </th>
##    <th style="text-align:right;"> mse_5% </th>
##    <th style="text-align:right;"> mse_10% </th>
##    <th style="text-align:right;"> se_0% </th>
##    <th style="text-align:right;"> se_5% </th>
##    <th style="text-align:right;"> se_10% </th>
##   </tr>
##  </thead>
## <tbody>
##   <tr>
##    <td style="text-align:right;"> 0 </td>
##    <td style="text-align:right;"> 0.0505 </td>
##    <td style="text-align:right;"> 0.0491 </td>
##    <td style="text-align:right;"> 0.0501 </td>
##    <td style="text-align:right;"> 0.00077 </td>
##    <td style="text-align:right;"> 0.00074 </td>
##    <td style="text-align:right;"> 0.00077 </td>
##   </tr>
```

```
##   <tr>
##    <td style="text-align:right;"> 1 </td>
##    <td style="text-align:right;"> 0.0521 </td>
##    <td style="text-align:right;"> 0.0517 </td>
##    <td style="text-align:right;"> 0.0528 </td>
##    <td style="text-align:right;"> 0.00066 </td>
##    <td style="text-align:right;"> 0.00065 </td>
##    <td style="text-align:right;"> 0.00065 </td>
##   </tr>
##   <tr>
##    <td style="text-align:right;"> 2 </td>
##    <td style="text-align:right;"> 0.0639 </td>
##    <td style="text-align:right;"> 0.0637 </td>
##    <td style="text-align:right;"> 0.0633 </td>
##    <td style="text-align:right;"> 0.00074 </td>
##    <td style="text-align:right;"> 0.00072 </td>
##    <td style="text-align:right;"> 0.00071 </td>
##   </tr>
##   <tr>
##    <td style="text-align:right;"> 3 </td>
##    <td style="text-align:right;"> 0.0756 </td>
##    <td style="text-align:right;"> 0.0771 </td>
##    <td style="text-align:right;"> 0.0782 </td>
##    <td style="text-align:right;"> 0.00082 </td>
##    <td style="text-align:right;"> 0.00083 </td>
##    <td style="text-align:right;"> 0.00083 </td>
##   </tr>
##   <tr>
##    <td style="text-align:right;"> 4 </td>
##    <td style="text-align:right;"> 0.0866 </td>
##    <td style="text-align:right;"> 0.0878 </td>
##    <td style="text-align:right;"> 0.0897 </td>
##    <td style="text-align:right;"> 0.00088 </td>
##    <td style="text-align:right;"> 0.00088 </td>
##    <td style="text-align:right;"> 0.00088 </td>
##   </tr>
##   <tr>
##    <td style="text-align:right;"> 5 </td>
##    <td style="text-align:right;"> 0.0999 </td>
##    <td style="text-align:right;"> 0.1001 </td>
##    <td style="text-align:right;"> 0.1020 </td>
##    <td style="text-align:right;"> 0.00096 </td>
##    <td style="text-align:right;"> 0.00096 </td>
##    <td style="text-align:right;"> 0.00096 </td>
##   </tr>
```

```
##   <tr>
##    <td style="text-align:right;"> 6 </td>
##    <td style="text-align:right;"> 0.1060 </td>
##    <td style="text-align:right;"> 0.1091 </td>
##    <td style="text-align:right;"> 0.1092 </td>
##    <td style="text-align:right;"> 0.00101 </td>
##    <td style="text-align:right;"> 0.00102 </td>
##    <td style="text-align:right;"> 0.00102 </td>
##   </tr>
##   <tr>
##    <td style="text-align:right;"> 7 </td>
##    <td style="text-align:right;"> 0.1147 </td>
##    <td style="text-align:right;"> 0.1184 </td>
##    <td style="text-align:right;"> 0.1223 </td>
##    <td style="text-align:right;"> 0.00106 </td>
##    <td style="text-align:right;"> 0.00108 </td>
##    <td style="text-align:right;"> 0.00109 </td>
##   </tr>
##   <tr>
##    <td style="text-align:right;"> 8 </td>
##    <td style="text-align:right;"> 0.1214 </td>
##    <td style="text-align:right;"> 0.1245 </td>
##    <td style="text-align:right;"> 0.1299 </td>
##    <td style="text-align:right;"> 0.00113 </td>
##    <td style="text-align:right;"> 0.00114 </td>
##    <td style="text-align:right;"> 0.00115 </td>
##   </tr>
##   <tr>
##    <td style="text-align:right;"> 9 </td>
##    <td style="text-align:right;"> 0.1259 </td>
##    <td style="text-align:right;"> 0.1315 </td>
##    <td style="text-align:right;"> 0.1360 </td>
##    <td style="text-align:right;"> 0.00117 </td>
##    <td style="text-align:right;"> 0.00119 </td>
##    <td style="text-align:right;"> 0.00120 </td>
##   </tr>
## </tbody>
## </table>
```

Part D: Results and Interpretation

Findings:

Hide

```
# Create tables and plots summarizing your results
```

Interpretation:

What did you learn?

```
I learned that no trimming is always better if you trim f
rom both sides in exponential becasue it is one sided.
```

Did the results surprise you?

```
Yes the results suprised me I thought some trimming would
help.
```

What are the practical implications?

```
For exponentially distributed data we should use one side
d trimming if we believe there is contamination.
```

Theory Connection:

Are there theoretical results that support/explain your findings?

```
Yes it is intuitive that it does not make sense to trim f
rom both sides when one sided distribution.
```

Reflection and Synthesis

Group Discussion Questions:

1. **Across all exercises, what's the key trade-off with trimmed means?**

   Your group's answer:

   ```
   It is ahrd to tell how much you should trim and timi
   ng too much can actually increase MSE.
   ```

2. **In practice, how would you choose a trim level without knowing the true contamination rate?**

   Ideas:

   ```
   I would do it based on expected contamination rate.
   ```

3. **What are the limitations of the contamination model we used?**

   Your thoughts:

   ```
   It doesnt work well for one sided distributions.
   ```

4. **How could you extend this methodology to real data analysis?**

   Applications:

   ```
   It would help purify real data sets to make more acc
   urate conclusions.
   ```

Bonus Challenges

If you finish early, try these extensions:

Challenge 1: Confidence Intervals

Compute confidence intervals for the MSE estimates and see if theoretical values fall within them.

Hide

```
# Template
compute_mse_ci <- function(estimates, true_value, conf_le
        vel = 0.95) {
  # YOUR CODE
}
```

Challenge 2: Power Analysis

How many MC iterations (m) do you need to reliably detect a 10% difference in MSE between two estimators?

Challenge 3: Bias-Variance Decomposition

Decompose MSE into bias² and variance for each estimator. Which dominates?

Hide

```
# Template
bias_variance_decomp <- function(estimates, true_value) {
  bias <- mean(estimates) - true_value
  variance <- var(estimates)

  list(
    bias_squared = bias^2,
    variance = variance,
    mse = bias^2 + variance
  )
}
```

Summary Presentation (3 minutes)

Prepare a brief presentation for the class:

Slide 1: Research Question

- What did you investigate?
- Why is it interesting?

Slide 2: Methods

- How did you design your MC study?
- What parameters did you test?

Slide 3: Key Results

- Show your main plot/table
- Highlight most interesting finding

Slide 4: Conclusions

- What did you learn?
- Practical implications?
- Connection to theory?

Additional Resources

Further Reading

1. **Robust Statistics:**
   - Huber, P. J. (1981). *Robust Statistics*
   - Hampel, F. R., et al. (1986). *Robust Statistics: The Approach Based on Influence Functions*
2. **Monte Carlo Methods:**
   - Rizzo, M. L. (2019). *Statistical Computing with R* - Chapter 6
   - Robert, C. P., & Casella, G. (2004). *Monte Carlo Statistical Methods*
3. **Real Applications:**
   - Deliveroo article: https://deliveroo.engineering/2018/12/07/monte-carlo-power-analysis.html (https://deliveroo.engineering/2018/12/07/monte-carlo-power-analysis.html)

R Packages to Explore

Hide

```
# Robust statistics
install.packages("robustbase")
install.packages("MASS")

# Visualization
install.packages("ggplot2")
install.packages("plotly")  # Interactive plots

# Power analysis
install.packages("pwr")
```

Submission Checklist

Before you finish, make sure you've:

- ☐ **Completed all three main exercises**
- ☐ **Designed and run your own study (Exercise 4)**
- ☐ **Interpreted all results (not just reported numbers)**
- ☐ **Connected findings to theoretical concepts**
- ☐ **Created clear visualizations**
- ☐ **Prepared your 3-minute presentation**
- ☐ **Discussed as a group and recorded insights**

Turn in: This completed worksheet + presentation slides

Good luck and have fun exploring! Remember: The goal isn't just to get the right answer, but to understand the process of using Monte Carlo methods to answer statistical questions.