Simulations of Trust Game

Your Name

2025-07-07

Simulations of Trust Game

In Berg's experiment, there does not seem to be a significant difference in the amount sent by participants in the treatment group (those with social history) compared to those without. Let Y_1 and Y_2 be two random variables associated to the control/baseline and the treatment group respectively. and $\mathcal{Z} \subset \mathcal{R}$ and $Y_1, Y_2 \in \mathcal{R}$. Furthermore, let P_{Y_1} and P_{Y_2} be the probability distributions of Y_1 and Y_2 respectively.

There is not statistical evidence to reject any of the following null hypothesis:

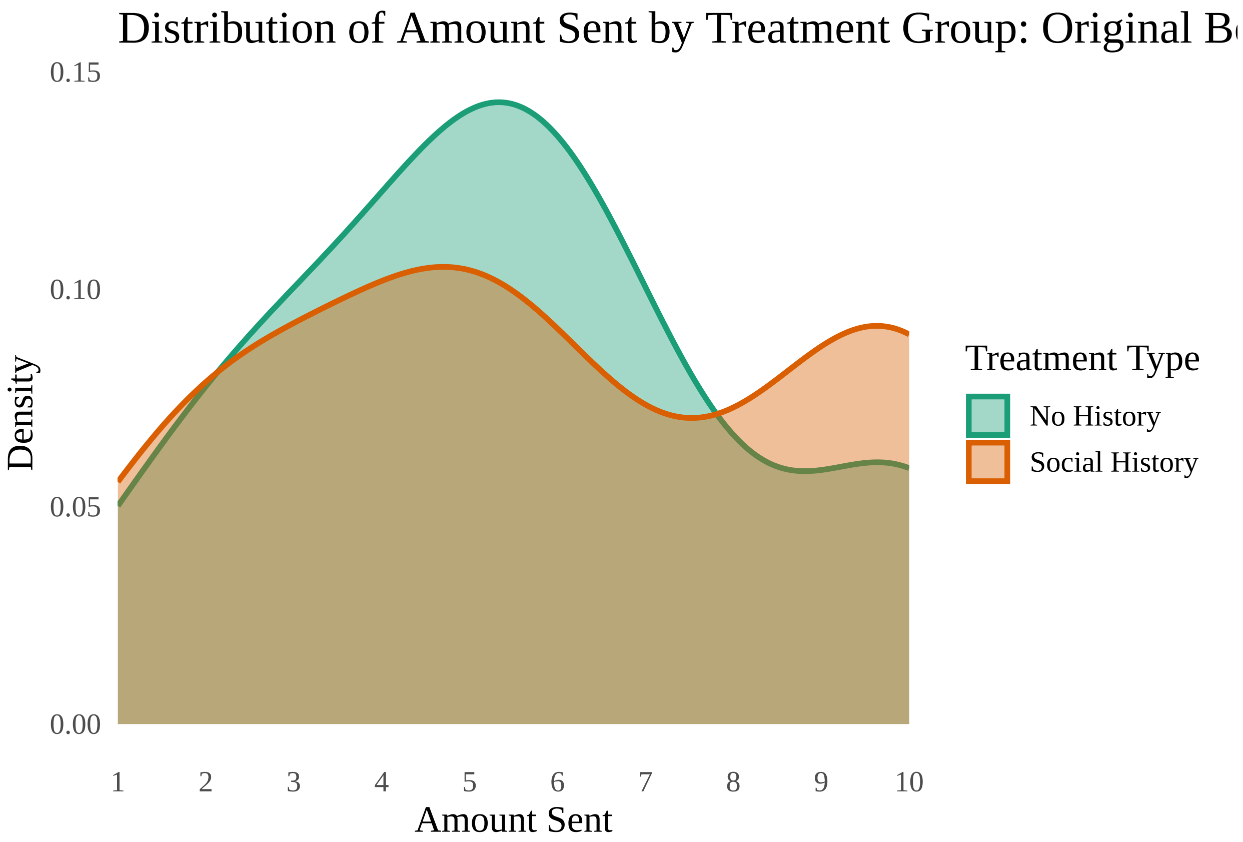
- H_0: The distribution of amounts sent is the same for both groups, i.e., $H_0 : Y_1 \overset{d}{=} Y_2$.
- H_0: The mean amount sent is the same for both groups, i.e., $\mathbb{E}[Y_1] = \mathbb{E}[Y_2]$.
- H_0: The probability that a randomly chosen individual from the treatment group sends more than one from the control group is 0.5, i.e., $P(Y_2 > Y_1) = P(Y_1 > Y_2) = 0.5$.

```r
# No output is produced by loading the ggplot2 library, so nothing will be printed.
library(exactRankTests)
library(ggplot2)
library(ggthemes)   # Optional for extra themes
library(scales)
library(progress)
library(knitr)
library(readxl)
library(dplyr)

source("si.test.R")

berg_1995 <- data.frame(
  treatment = c(
    rep(0, 32),  # No History
    rep(1, 28)   # Social History
  ),
  sent = c(
    7, 3, 7, 5, 3, 2, 6, 4, 10, 5, 5, 10, 5, 8, 5, 0, 7, 1, 3, 2, 6, 10, 3, 6, 6, 10, 10, 6, 4, 0, 5, 1
    5, 10, 2, 5, 0, 2, 10, 10, 5, 9, 3, 2, 10, 2, 5, 5, 8, 5, 0, 1, 10, 7, 10, 3, 10, 6, 5, 0
  ),
  sent_back = c(
    1, 0, 6, 11, 1, 4, 0, 1, 20, 5, 7, 0, 0, 4, 15, 0, 1, 0, 5, 0, 12, 15, 6, 8, 1, 15, 1, 3, 1, 0, 5,
    11, 15, 2, 8, 0, 1, 16, 15, 5, 0, 0, 0, 5, 0, 0, 10, 3, 8, 0, 1, 20, 14, 15, 6, 10, 8, 8, 0
  )
)

# Compute mean and standard deviation by treatment and print
library(dplyr)
```

```r
berg_1995 %>%
  group_by(treatment) %>%
  summarise(
  mean_sent = mean(sent),
  sd_sent = sd(sent)
  ) %>%
  print()
```

```
## # A tibble: 2 x 3
##   treatment mean_sent sd_sent
##       <dbl>     <dbl>   <dbl>
## 1         0      5.16    2.94
## 2         1      5.36    3.53
```

```r
palette <- c("#1b9e77", "#d95f02")

ggplot(
  berg_1995,
  aes(
    x = sent,
    fill = factor(
      treatment,
      labels = c("No History", "Social History")
    ),
    color = factor(
      treatment,
      labels = c("No History", "Social History")
    )
  )
) +
  geom_density(alpha = 0.4, size = 1) +
  scale_fill_manual(values = palette) +
  scale_color_manual(values = palette) +
  scale_x_continuous(
    breaks = 1:10,
    limits = c(1, 10),
    expand = c(0, 0)
  ) +
  labs(
    title = "Distribution of Amount Sent by Treatment Group: Original Berg (1995) Data",
    x = "Amount Sent",
    y = "Density",
    fill = "Treatment Type",
    color = "Treatment Type"
  ) +
  theme_minimal(base_size = 14, base_family = "Times") +
  theme(panel.grid = element_blank())
```



```
# Perform Wilcoxon rank-sum test (Mann-Whitney U) to compare amount sent between groups
wilcox_test_result <- wilcox.exact(sent ~ treatment, data = berg_1995, alternative = "two.sided" )
t_test_result <- t.test(sent ~ treatment, data = berg_1995, alternative = "greater")
# Check what is that we are tresting here
group0 <- berg_1995$sent[
  berg_1995$treatment == 0
]
group2 <- berg_1995$sent[
  berg_1995$treatment == 1
]
if (length(group0) > 0 && length(group2) > 0) {
  si_test_result <- si.test(group0, group2)
} else {
  si_test_result <- NA
}

# Print results of Wilcoxon and t-test in an organized way
# Create a data frame with the main results for LaTeX table
results_table <- data.frame(
  Test = c("Wilcoxon Exact", "T-Test", "Stochastic Inequality"),
  Statistic = c(
    round(wilcox_test_result$statistic, 3),
    round(t_test_result$statistic, 3),
    if (!is.null(si_test_result$statistic)) round(si_test_result$statistic, 3) else NA
  ),
  P_Value = c(
```

Table 1: Summary of Statistical Test Results

	Test	Statistic	P_Value	Alternative
W	Wilcoxon Exact	444.000	0.956	two.sided
t	T-Test	-0.238	0.593	greater
	Stochastic Inequality	NA	1.000	two.sided

```
    signif(wilcox_test_result$p.value, 3),
    signif(t_test_result$p.value, 3),
    if (!is.null(si_test_result$p.value)) signif(si_test_result$p.value, 3) else NA
  ),
  Alternative = c(
    wilcox_test_result$alternative,
    t_test_result$alternative,
    if (!is.null(si_test_result$alternative)) si_test_result$alternative else NA
  )
)

# Print as LaTeX table
knitr::kable(
  results_table,
  format = "latex",
  booktabs = TRUE,
  caption = "Summary of Statistical Test Results"
)
```

Simulations of the Trust Game

1.1 Size

We ran 1000 simulations for the treatment and the control group to evaluate the power of the tests. In other words, if our treatment and control were to generate a sample from the same distribution. (Knowing that there is no evidence agains the null hypothesis, that they come from the same distribution). We would expect that on average in less tha 5% of the cases, we will find evidence to reject the null hypothesis, meaning that the p-values are greater than 0.05. (All our tests have the same level)

```
#  SIMULATION 1

reps <- 1000
n <- 100

# File paths for cached results
t_pvalues_file <- "t_pvalues_sim1.rds"
wilcox_pvalues_file <- "wilcox_pvalues_sim1.rds"

# Check if cached results exist
if (file.exists(t_pvalues_file) && file.exists(wilcox_pvalues_file)) {
  t_pvalues <- readRDS(t_pvalues_file)
  wilcox_exact_pvalues <- readRDS(wilcox_pvalues_file)
} else {
  t_pvalues <- numeric(reps)
```

Table 2: Estimated Power for t-test and Wilcoxon Test (p-value < 0.05)

Test	Power
t-test	0.075
Wilcoxon Exact	0.055

```
  wilcox_exact_pvalues <- numeric(reps)
  for (i in 1:reps) {
    control <- sample(berg_1995$sent[berg_1995$treatment == 0], size = n, replace = TRUE)
    treatment <- sample(berg_1995$sent[berg_1995$treatment == 1], size = n, replace = TRUE)
    t_pvalues[i] <- t.test(control, treatment)$p.value
    wilcox_exact_pvalues[i] <- wilcox.exact(control, treatment)$p.value
  }
  saveRDS(t_pvalues, t_pvalues_file)
  saveRDS(wilcox_exact_pvalues, wilcox_pvalues_file)
}

t_test_power <- mean(t_pvalues < 0.05)
wilcox_power <- mean(wilcox_exact_pvalues < 0.05)

# Create results table
power_table <- data.frame(
  Test = c("t-test", "Wilcoxon Exact"),
  Power = c(round(t_test_power, 3), round(wilcox_power, 3))
)

# Print as LaTeX table
knitr::kable(
  power_table,
  format = "latex",
  booktabs = TRUE,
  caption = "Estimated Power for t-test and Wilcoxon Test (p-value < 0.05)"
)
```

1.2 Power

Now, assuming that the treatment indeed came from a different distribution, that gives the following probabilities of sending different amounts, so that:

weights_llms <- c(“0” = 0.03, “1” = 0.04, “2” = 0.05, “3” = 0.6, “4” = 0.07, “5” = 0.25, “6” = 0.5, “7” = 0.025, “8” = 0.1, “9” = 0.025, “10” = 0.3)

We are exharcebating the effects observed in Berg (1995) treatment, so that the treatment group is more likely to send 6, and 10, and less likely to send 1, compared to the control group. By sampling from such distribution we will know in advance that under the alternative hypothesis, we will find a significant difference between the two groups, and we can estimate the power of the tests under the alternative hypothesis. We start by running the simulation once and observe the results of the tests, and then we will run the simulations for different sample sizes, to estimate the power of the tests under the alternative hypothesis.

```
# SIMULATION 2

# File paths for cached results
llm_means_file <- "means_mat_llm_sim2.rds"
```

```r
llm_medians_file <- "medians_mat_llm_sim2.rds"
llm_t_pvalues_file <- "t_pvalues_llm_sim2.rds"
llm_wilcox_pvalues_file <- "wilcox_pvalues_llm_sim2.rds"
llm_si_pvalues_file <- "si_pvalues_llm_sim2.rds"

# Compute frequency probabilities for each group in berg_1995$sent
sent_no_treatment <- berg_1995$sent[berg_1995$treatment == 0]
sent_social_treatment <- berg_1995$sent[berg_1995$treatment == 1]

# Get unique values
unique_sent <- sort(unique(berg_1995$sent))

# Compute probabilities for each value in each group
weights_no_treatment <- round(as.numeric(table(factor(sent_no_treatment, levels = unique_sent)) / lengt
weights_social_treatment <- round(as.numeric(table(factor(sent_social_treatment, levels = unique_sent))

weights_db <- data.frame(
  sent = unique_sent,
  weight_no_treatment = weights_no_treatment,
  weight_social_treatment = weights_social_treatment
)

# Participants in the Treatment condition in Berg's experiment were more likely to send 10, and less li

print(weights_no_treatment)
```

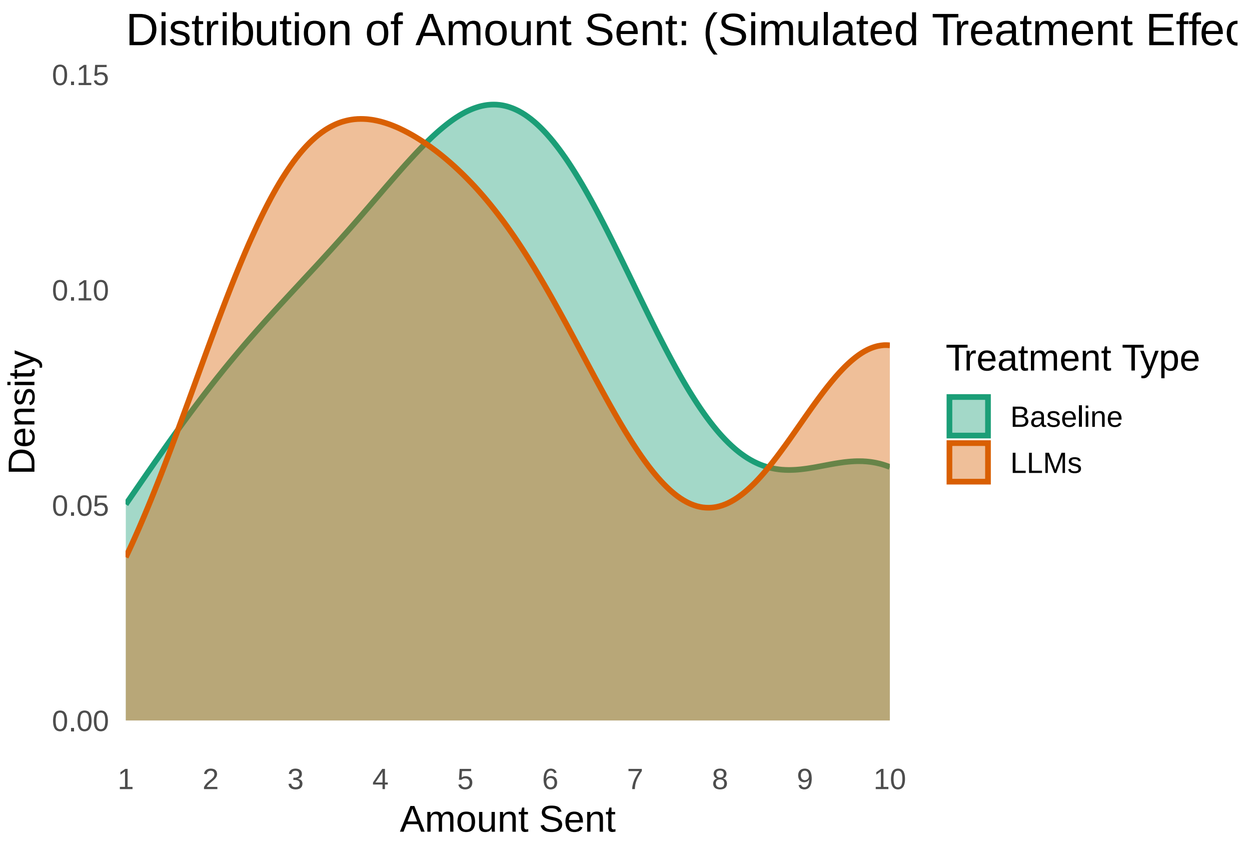
```
##  [1] 0.06 0.06 0.06 0.12 0.06 0.19 0.16 0.09 0.03 0.00 0.16
```

```r
print(weights_social_treatment)
```

```
##  [1] 0.11 0.04 0.14 0.07 0.00 0.25 0.04 0.04 0.04 0.04 0.25
```

```r
# Assuming that the effect on our treatment will increase the probability of giving 6 , and 10, and red
# of giving 1, we can create a vector of weights that will reflect this.

weights_llms <- c(0.03, 0.04, 0.05, 0.6, 0.07, 0.25, 0.5, 0.025, 0.1, 0.025, 0.3)

treatment_llms <- sample(
  unique_sent,
  size = dim(berg_1995[berg_1995$treatment == 0, ])[1],
  replace = TRUE,
  prob = weights_llms
)

# Adding to the data frame, and testing for possible effects, with the distribution above.

berg_1995_v1 <- rbind(
  berg_1995,
  data.frame(
    treatment = rep(2, dim(berg_1995[berg_1995$treatment == 0, ])[1]),
    sent = treatment_llms,
    sent_back = NA  # Assuming no response for the treatment group
  )
)
```

```r
berg_1995_fake_llm_treatment <- berg_1995_v1[berg_1995_v1$treatment != 1, ]


ggplot(
  berg_1995_fake_llm_treatment,
  aes(
  x = sent,
  fill = factor(
    treatment,
    labels = c("Baseline", "LLMs")
  ),
  color = factor(
    treatment,
    labels = c("Baseline", "LLMs")
  )
  )
) +
  geom_density(alpha = 0.4, size = 1) +
  scale_fill_manual(values = palette) +
  scale_color_manual(values = palette) +
  scale_x_continuous(
  breaks = 1:10,
  limits = c(1, 10),
  expand = c(0, 0)
  ) +
  labs(
  title = "Distribution of Amount Sent: (Simulated Treatment Effects)",
  x = "Amount Sent",
  y = "Density",
  fill = "Treatment Type",
  color = "Treatment Type"
  ) +
theme_minimal(base_size = 14) + theme(panel.grid = element_blank())
```



```r
wilcox_test_result <- wilcox.exact(sent ~ treatment, data = berg_1995_fake_llm_treatment, alternative =
t_test_result <- t.test(sent ~ treatment, data = berg_1995_fake_llm_treatment, alternative = "two.sided

print(wilcox_test_result)
```

```
##
##  Exact Wilcoxon rank sum test
##
## data:  sent by treatment
## W = 466, p-value = 0.5352
## alternative hypothesis: true mu is not equal to 0
```

```r
print(t_test_result)
```

```
##
##  Welch Two Sample t-test
##
## data:  sent by treatment
## t = -0.94196, df = 61.986, p-value = 0.3499
## alternative hypothesis: true difference in means between group 0 and group 2 is not equal to 0
## 95 percent confidence interval:
##  -2.1464808  0.7714808
## sample estimates:
## mean in group 0 mean in group 2
##         5.15625         5.84375
```

```r
# Power Results for each test
# Power analysis for t-test, Wilcoxon, and Stochastic Inequality test using the simulated LLM treatment
```

```
if (
  file.exists(llm_t_pvalues_file) &&
  file.exists(llm_wilcox_pvalues_file) &&
  file.exists(llm_si_pvalues_file) &&
  file.exists(llm_means_file) &&
  file.exists(llm_medians_file)
) {
  t_pvalues <- readRDS(llm_t_pvalues_file)
  wilcox_pvalues <- readRDS(llm_wilcox_pvalues_file)
  si_pvalues <- readRDS(llm_si_pvalues_file)
  means_mat <- readRDS(llm_means_file)
  medians_mat <- readRDS(llm_medians_file)
} else {
  t_pvalues <- numeric(reps)
  wilcox_pvalues <- numeric(reps)
  si_pvalues <- numeric(reps)
  means_mat <- matrix(NA, nrow = reps, ncol = 2)
  medians_mat <- matrix(NA, nrow = reps, ncol = 2)

  pb <- progress_bar$new(
  format = "  Simulating [:bar] :percent eta: :eta",
  total = reps, clear = FALSE, width = 60
  )

  for (i in 1:reps) {
  control <- sample(berg_1995$sent[berg_1995$treatment == 0], size = n, replace = TRUE)
  treatment <- sample(unique_sent, size = n, replace = TRUE, prob = weights_llms)

  # t-test
  t_pvalues[i] <- t.test(control, treatment)$p.value

  # Wilcoxon exact test
  wilcox_pvalues[i] <- wilcox.exact(control, treatment)$p.value

  # Stochastic Inequality test
  si_res <- si.test(control, treatment)
  si_pvalues[i] <- if (!is.null(si_res$p.value)) si_res$p.value else NA

  # Store means and medians for this simulation
  means_mat[i, ] <- c(mean(control), mean(treatment))
  medians_mat[i, ] <- c(median(control), median(treatment))

  pb$tick()
  }

  saveRDS(t_pvalues, llm_t_pvalues_file)
  saveRDS(wilcox_pvalues, llm_wilcox_pvalues_file)
  saveRDS(si_pvalues, llm_si_pvalues_file)
  saveRDS(means_mat, llm_means_file)
  saveRDS(medians_mat, llm_medians_file)
}

# Compute average means and medians across all simulations
```

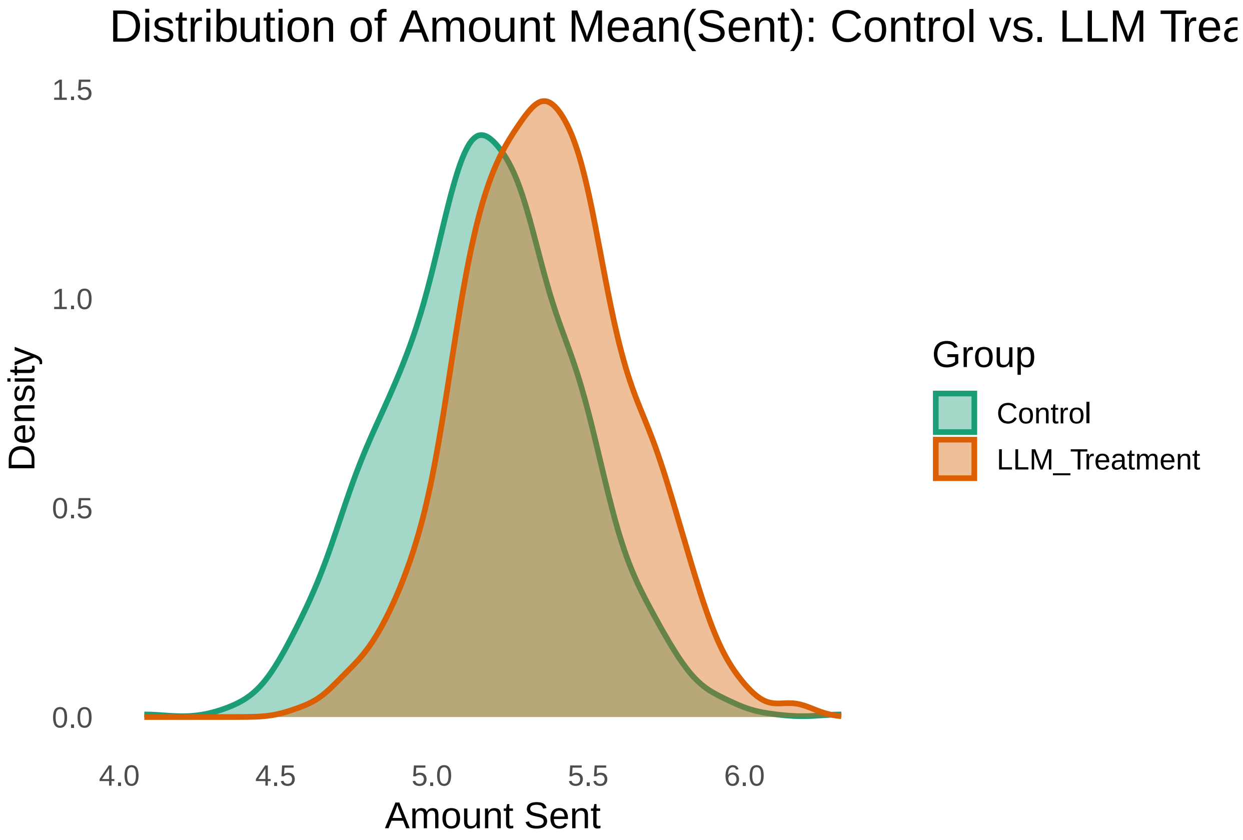
```r
summary_stats <- data.frame(
  Group = c("Control", "LLM Treatment"),
  Mean = round(colMeans(means_mat, na.rm = TRUE), 3),
  Median = round(colMeans(medians_mat, na.rm = TRUE), 3)
)
print(summary_stats)
```

```
##           Group  Mean Median
## 1       Control 5.160  5.101
## 2 LLM Treatment 5.359  5.318
```

```r
# Transform means_mat for plotting
means_mat <- as.data.frame(means_mat)
colnames(means_mat) <- c("Control", "LLM_Treatment")
means_mat_long <- tidyr::pivot_longer(
  means_mat,
  cols = everything(),
  names_to = "group",
  values_to = "value"
)

library(ggplot2)
ggplot(means_mat_long, aes(x = value, fill = group, color = group)) +
  geom_density(alpha = 0.4, size = 1) +
  scale_fill_manual(values = c("#1b9e77", "#d95f02")) +
  scale_color_manual(values = c("#1b9e77", "#d95f02")) +
  labs(
  title = "Distribution of Amount Mean(Sent): Control vs. LLM Treatment (Simulated)",
  x = "Amount Sent",
  y = "Density",
  fill = "Group",
  color = "Group"
  ) +
  theme_minimal(base_size = 14) +
  theme(panel.grid = element_blank())
```

Table 3: Estimated Power for t-test, Wilcoxon, and Stochastic Inequality Test (Simulated LLM Treatment, p-value < 0.05)

Test	Power
t-test	0.087
Wilcoxon Exact	0.069
Stochastic Inequality	0.023



```
power_table_llm <- data.frame(
  Test = c("t-test", "Wilcoxon Exact", "Stochastic Inequality"),
  Power = c(
  round(mean(t_pvalues < 0.05, na.rm = TRUE), 3),
  round(mean(wilcox_pvalues < 0.05, na.rm = TRUE), 3),
  round(mean(si_pvalues < 0.05, na.rm = TRUE), 3)
  )
)

knitr::kable(
  power_table_llm,
  format = "latex",
  booktabs = TRUE,
  caption = "Estimated Power for t-test, Wilcoxon, and Stochastic Inequality Test (Simulated LLM Treatm
)
```

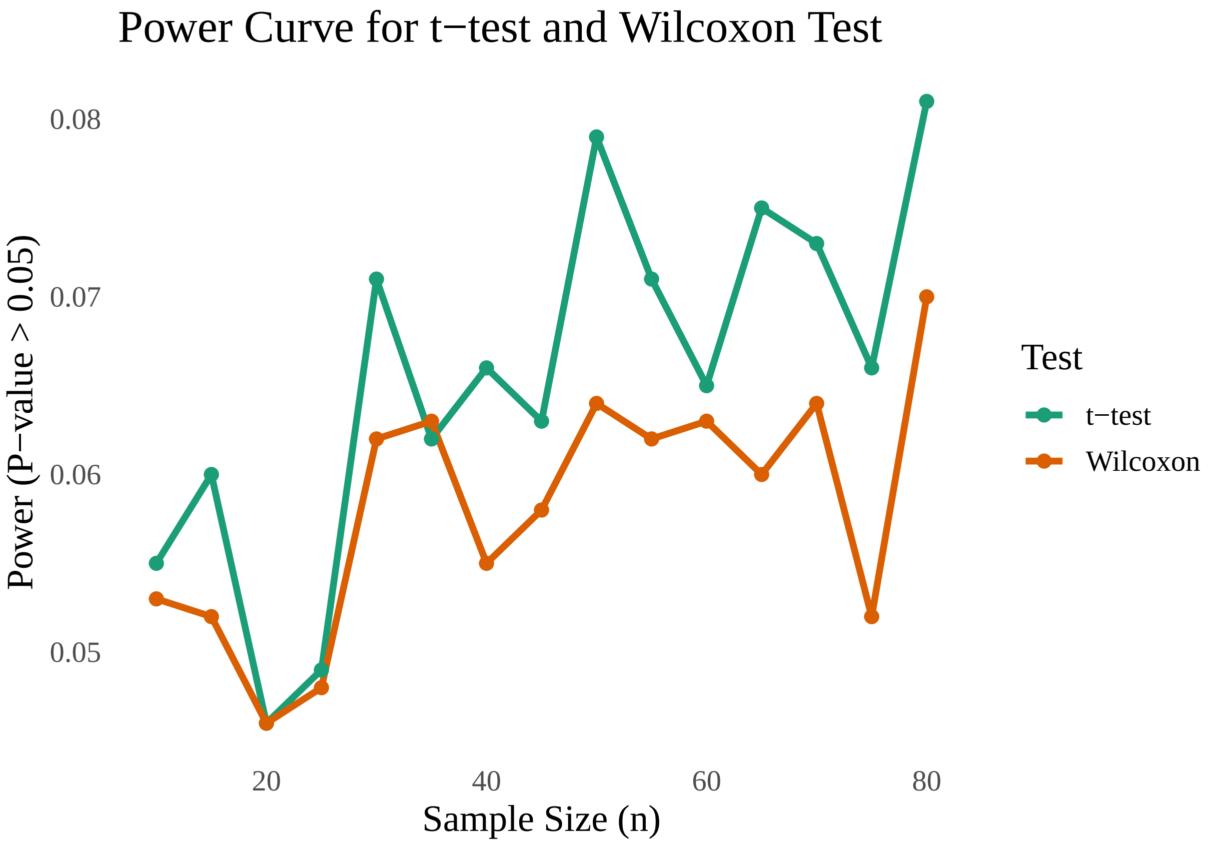
1.3 Power Analysis

We will now compare the power of the three tests (t-test, Wilcoxon, and Stochastic Inequality test) to determine how many samples we need to detect a significant difference between the two groups, assuming the treatment group has a different distribution of amounts sent.

```r
# SIMULATION 3
# Define a sequence of n values
n_values <- seq(10, 80, by = 5)

# File paths for cached results
power_results_file <- "power_results_sim3.rds"

# Checking if there is any difference in means:
# Difference in means of approximately 0.75 points. (or 7.5%)
aggregate(sent ~ treatment, data = berg_1995, mean)
```

```
##   treatment     sent
## 1         0 5.156250
## 2         1 5.357143
```

```r
if (file.exists(power_results_file)) {
  power_results <- readRDS(power_results_file)
} else {
  # Prepare data frame to store results
  power_results <- data.frame(
    n = integer(),
    t_test_power = numeric(),
    wilcox_power = numeric()
  )

  pb <- progress_bar$new(
    format = "  Simulating [:bar] :percent eta: :eta",
    total = length(n_values), clear = FALSE, width = 60
  )
  for (n in n_values) {
    # t-test power
    t_pvalues <- numeric(reps)
    w_pvalues <- numeric(reps)
    for (i in 1:reps) {
      control <- sample(berg_1995$sent[berg_1995$treatment == 0], size = n, replace = TRUE)
      treatment <- sample(unique_sent, size = n, replace = TRUE, prob = weights_llms)
      t_pvalues[i] <- t.test(control, treatment)$p.value
      w_pvalues[i] <- wilcox.exact(control, treatment)$p.value
    }
    t_test_power <- mean(t_pvalues < 0.05)
    wilcox_power <- mean(w_pvalues < 0.05)
    # Store results
    power_results <- rbind(
      power_results,
      data.frame(n = n, t_test_power = t_test_power, wilcox_power = wilcox_power)
    )
    pb$tick()
  }
  saveRDS(power_results, power_results_file)
}

# Reshape for plotting
library(tidyr)
power_long <- pivot_longer(power_results, cols = c("t_test_power", "wilcox_power"),
                           names_to = "test", values_to = "power")

# Plot
library(ggplot2)
ggplot(power_long, aes(x = n, y = power, color = test)) +
  geom_line(size = 1.2) +
  geom_point(size = 2) +
  labs(
    title = "Power Curve for t-test and Wilcoxon Test",
    x = "Sample Size (n)",
    y = "Power (P-value > 0.05)",
    color = "Test"
  ) +
  scale_color_manual(values = c("t_test_power" = "#1b9e77", "wilcox_power" = "#d95f02"),
                     labels = c("t_test_power" = "t-test", "wilcox_power" = "Wilcoxon")) +
  theme_minimal(base_size = 14, base_family = "Times") +
  theme(panel.grid = element_blank())
```



```
# Print the power results as a LaTeX table
knitr::kable(
  power_results,
```

Table 4: Power Results for t-test and Wilcoxon Test Across Sample Sizes

n	t_test_power	wilcox_power
10	0.055	0.053
15	0.060	0.052
20	0.046	0.046
25	0.049	0.048
30	0.071	0.062
35	0.062	0.063
40	0.066	0.055
45	0.063	0.058
50	0.079	0.064
55	0.071	0.062
60	0.065	0.063
65	0.075	0.060
70	0.073	0.064
75	0.066	0.052
80	0.081	0.070

```
  format = "latex",
  booktabs = TRUE,
  caption = "Power Results for t-test and Wilcoxon Test Across Sample Sizes"
)
```

1.4 Including Interaction Effects

We expect that users that are more familiarized with LLMs, will be more likely to send higher amounts. We will then simply ask participants to provide us with their email address with which they are registered in OpenAI

```
# SIMULATION 4
# Simulate interaction: probability of having an LLM account is higher in the treatment group

set.seed(123) # for reproducibility

# File paths for cached results
reg_pvalues_int_file <- "reg_pvalues_int_sim4.rds"
wilcox_pvalues_by_llm_file <- "wilcox_pvalues_by_llm_sim4.rds"

if (file.exists(reg_pvalues_int_file) && file.exists(wilcox_pvalues_by_llm_file)) {
  reg_pvalues_int <- readRDS(reg_pvalues_int_file)
  wilcox_pvalues_by_llm <- readRDS(wilcox_pvalues_by_llm_file)
} else {
  reg_pvalues_int <- data.frame(
    group_treatment = numeric(reps),
    llm_account = numeric(reps),
    interaction = numeric(reps)
  )

  wilcox_pvalues_by_llm <- data.frame(
```

```r
    rep = 1:reps,
    llm0 = NA_real_,
    llm1 = NA_real_
  )

  pb <- progress_bar$new(
    format = "  Simulating [:bar] :percent eta: :eta",
    total = reps, clear = FALSE, width = 60
  )

  for (i in 1:reps) {
    # Sample control group (baseline)
    control_sent <- sample(berg_1995$sent[berg_1995$treatment == 0], size = n, replace = TRUE)
    # Probability of LLM account increases with sent amount (e.g., logistic or linear scaling)
    control_llm_prob <- scales::rescale(control_sent, to = c(0.2, 0.7)) # adjust range as needed
    control_llm_account <- rbinom(n, 1, prob = control_llm_prob)

    # Sample treatment group (LLM treatment)
    treatment_sent <- sample(unique_sent, size = n, replace = TRUE, prob = weights_llms)
    treatment_llm_prob <- scales::rescale(treatment_sent, to = c(0.4, 0.95)) # higher base probability
    treatment_llm_account <- rbinom(n, 1, prob = treatment_llm_prob)

    # Combine into a single data frame
    sim_data <- data.frame(
      group = rep(c("control", "treatment"), each = n),
      sent = c(control_sent, treatment_sent),
      llm_account = c(control_llm_account, treatment_llm_account)
    )

    # Fit linear regression: sent ~ group + llm_account + group:llm_account
    sim_data$group <- factor(sim_data$group, levels = c("control", "treatment"))
    fit <- lm(sent ~ group + llm_account + group:llm_account, data = sim_data)
    reg_summary <- summary(fit)$coefficients

    # Extract p-values for group_treatment, llm_account, and interaction
    reg_pvalues_int$group_treatment[i] <- if ("grouptreatment" %in% rownames(reg_summary)) reg_summary[
    reg_pvalues_int$llm_account[i] <- if ("llm_account" %in% rownames(reg_summary)) reg_summary["llm_ac
    reg_pvalues_int$interaction[i] <- if ("grouptreatment:llm_account" %in% rownames(reg_summary)) reg_s

    # Wilcoxon test between control and treatment groups, separately for each llm_account value
    for (llm_val in c(0, 1)) {
      subset_data <- sim_data[sim_data$llm_account == llm_val, ]
      if (length(unique(subset_data$group)) == 2) {
        wilcox_res <- wilcox.exact(sent ~ group, data = subset_data)
        wilcox_pvalues_by_llm[i, paste0("llm", llm_val)] <- wilcox_res$p.value
      }
    }
    pb$tick()
  }

  saveRDS(reg_pvalues_int, reg_pvalues_int_file)
  saveRDS(wilcox_pvalues_by_llm, wilcox_pvalues_by_llm_file)
}
```

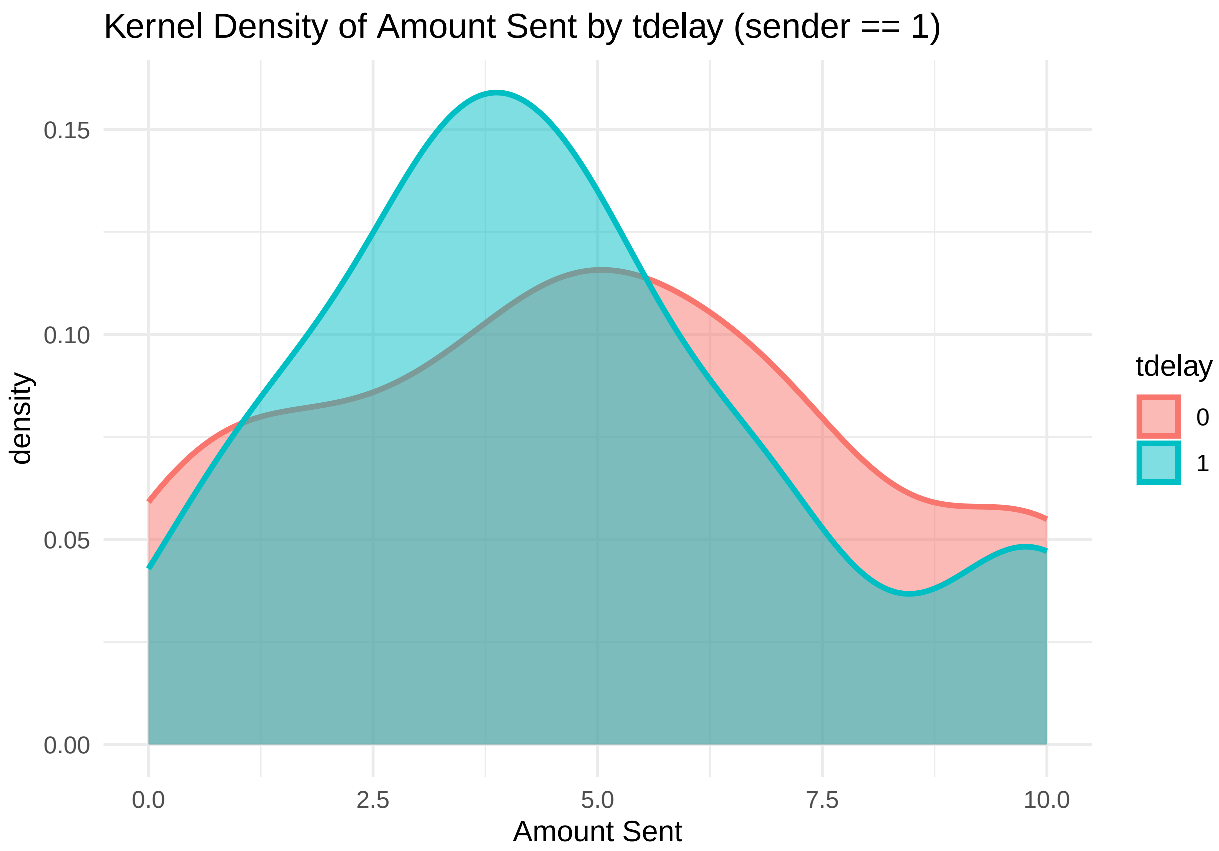
```
# Power of the tests
power_results_int <- data.frame(
  Test = c(
    "t-test (group)",
    "t-test (interaction)",
    "Wilcoxon Exact (llm_account=0)",
    "Wilcoxon Exact (llm_account=1)"
  ),
  Power = c(
    round(mean(reg_pvalues_int$group_treatment < 0.05, na.rm = TRUE), 3),
    round(mean(reg_pvalues_int$interaction < 0.05, na.rm = TRUE), 3),
    round(mean(wilcox_pvalues_by_llm$llm0 < 0.05, na.rm = TRUE), 3),
    round(mean(wilcox_pvalues_by_llm$llm1 < 0.05, na.rm = TRUE), 3)
  )
)

print(power_results_int)
```

```
##                             Test Power
## 1                 t-test (group) 0.043
## 2           t-test (interaction) 0.047
## 3 Wilcoxon Exact (llm_account=0) 0.049
## 4 Wilcoxon Exact (llm_account=1) 0.080
```

1.5 Newer Version Trust Game

Using data from the experiment in Nature (https://www.nature.com/articles/s41598-022-15420-2#Sec15)

```
# Load the Excel file (first sheet by default)
data_trustor_disregard <- read_excel("Data/Study1.xlsx")

# Subset data where sender == 1
subset <- subset(data_trustor_disregard, sender == 1)

ggplot(subset, aes(x = amountsent, fill = factor(tdelay), color = factor(tdelay))) +
  geom_density(alpha = 0.5, size = 1) +
  labs(
    title = "Kernel Density of Amount Sent by tdelay (sender == 1)",
    x = "Amount Sent",
    fill = "tdelay",
    color = "tdelay"
  ) +
  theme_minimal()
```



```r
# Compute and print table with means of amountsent by tdelay group
means_table <- subset %>%
  group_by(tdelay) %>%
  summarise(
    n = n(),
    mean_amountsent = mean(amountsent, na.rm = TRUE)
  )

print(means_table)
```

```
## # A tibble: 2 x 3
##   tdelay     n mean_amountsent
##    <dbl> <int>           <dbl>
## 1      0    75            4.92
## 2      1    75            4.54
```

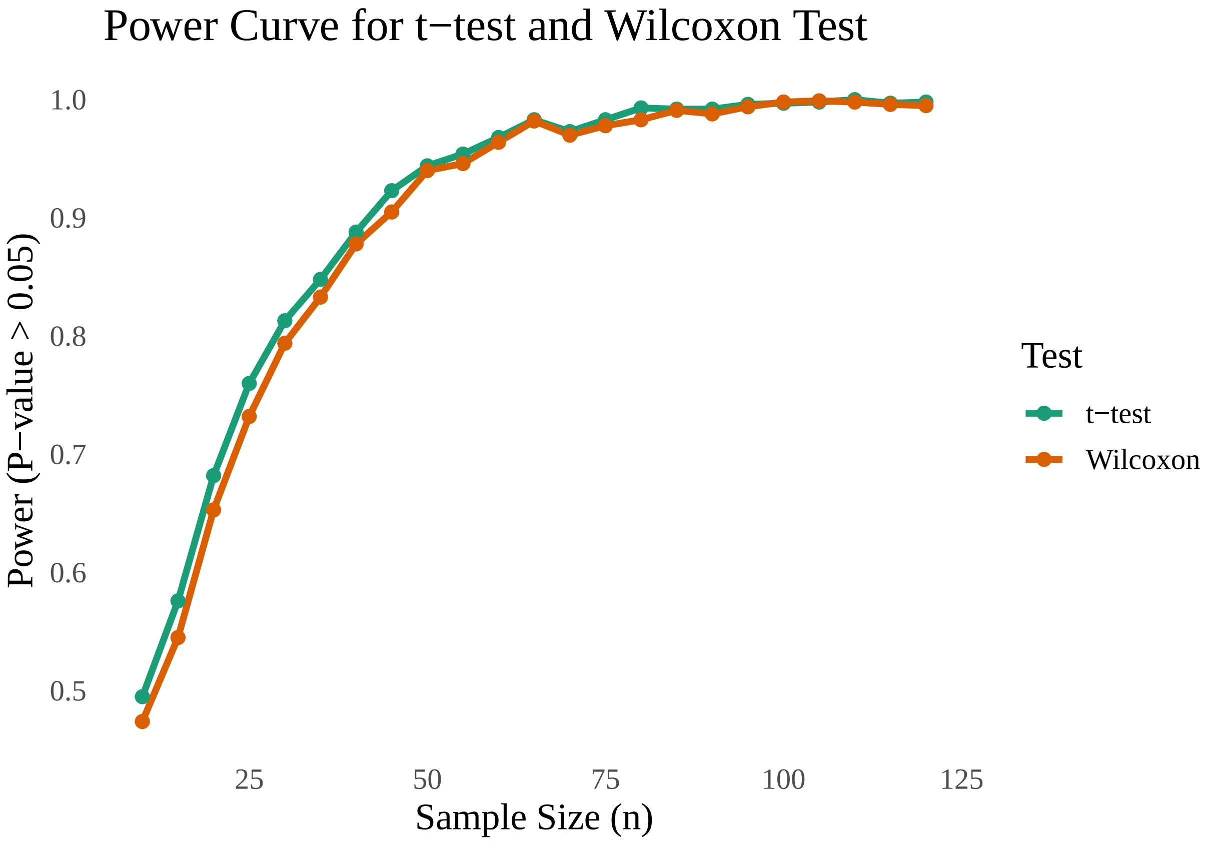
1.6 Power Analysis assuming continuous disrtritbution

We will now sample from a continuous distribution, and assume that the treatment group is more likely to send higher amounts, and the control group is more likely to send lower amounts. We assume that the design effect is of 0.5/

```r
# SIMULATION 5
# Define a sequence of n values
n_values <- seq(10, 120, by = 5)
mean_control <- 5
```

```r
mean_treatment <- 7.5

# File paths for cached results
power_results_file <- "power_results_sim5.rds"

if (file.exists(power_results_file)) {
  power_results <- readRDS(power_results_file)
} else {
  # Prepare data frame to store results
  power_results <- data.frame(
    n = integer(),
    t_test_power = numeric(),
    wilcox_power = numeric()
  )

  pb <- progress_bar$new(
    format = "  Simulating [:bar] :percent eta: :eta",
    total = length(n_values), clear = FALSE, width = 60
  )
  for (n in n_values) {
    # t-test power
    t_pvalues <- numeric(reps)
    w_pvalues <- numeric(reps)
    for (i in 1:reps) {
      control <- sample(
        rnorm(n, mean = mean_control, sd = 3),
        size = n,
        replace = TRUE
      )
      treatment <- sample(
        rnorm(n, mean = mean_treatment, sd = 3),
        size = n,
        replace = TRUE
      )
      t_pvalues[i] <- t.test(control, treatment)$p.value
      w_pvalues[i] <- wilcox.exact(control, treatment)$p.value
    }
    t_test_power <- mean(t_pvalues < 0.05)
    wilcox_power <- mean(w_pvalues < 0.05)
    # Store results
    power_results <- rbind(
      power_results,
      data.frame(n = n, t_test_power = t_test_power, wilcox_power = wilcox_power)
    )
    pb$tick()
  }
  saveRDS(power_results, power_results_file)
}

# Reshape for plotting
library(tidyr)
power_long <- pivot_longer(power_results, cols = c("t_test_power", "wilcox_power"),
                           names_to = "test", values_to = "power")
```

```r
# Plot
library(ggplot2)
ggplot(power_long, aes(x = n, y = power, color = test)) +
  geom_line(size = 1.2) +
  geom_point(size = 2) +
  labs(
    title = "Power Curve for t-test and Wilcoxon Test",
    x = "Sample Size (n)",
    y = "Power (P-value > 0.05)",
    color = "Test"
  ) +
  scale_color_manual(values = c("t_test_power" = "#1b9e77", "wilcox_power" = "#d95f02"),
                     labels = c("t_test_power" = "t-test", "wilcox_power" = "Wilcoxon")) +
  theme_minimal(base_size = 14, base_family = "Times") +
  theme(panel.grid = element_blank())
```



```r
# Print the power results as a LaTeX table
knitr::kable(
  power_results,
  format = "latex",
  booktabs = TRUE,
  caption = "Power Results for t-test and Wilcoxon Test Across Sample Sizes"
)
```

```r
# SIMULATION 6
# Simulate interaction: probability of having an LLM account is higher in the treatment group
```

Table 5: Power Results for t-test and Wilcoxon Test Across Sample Sizes

n	t_test_power	wilcox_power
10	0.495	0.474
15	0.576	0.545
20	0.682	0.653
25	0.760	0.732
30	0.813	0.794
35	0.848	0.833
40	0.888	0.878
45	0.923	0.905
50	0.944	0.940
55	0.954	0.946
60	0.968	0.964
65	0.983	0.982
70	0.973	0.970
75	0.983	0.978
80	0.993	0.983
85	0.992	0.991
90	0.992	0.988
95	0.996	0.994
100	0.997	0.998
105	0.998	0.999
110	1.000	0.998
115	0.997	0.996
120	0.998	0.995

```r
set.seed(123) # for reproducibility
n = 100
mean_control <- 5
mean_treatment <- 7.5

# File paths for cached results
reg_pvalues_int_file <- "reg_pvalues_int_sim6.rds"
wilcox_pvalues_by_llm_file <- "wilcox_pvalues_by_llm_sim6.rds"

if (file.exists(reg_pvalues_int_file) && file.exists(wilcox_pvalues_by_llm_file)) {
  reg_pvalues_int <- readRDS(reg_pvalues_int_file)
  wilcox_pvalues_by_llm <- readRDS(wilcox_pvalues_by_llm_file)
} else {
  reg_pvalues_int <- data.frame(
    group_treatment = numeric(reps),
    llm_account = numeric(reps),
    interaction = numeric(reps)
  )

  wilcox_pvalues_by_llm <- data.frame(
    rep = 1:reps,
    llm0 = NA_real_,
    llm1 = NA_real_
  )

  pb <- progress_bar$new(
    format = "  Simulating [:bar] :percent eta: :eta",
    total = reps, clear = FALSE, width = 60
  )
  for (i in 1:reps) {
    # Sample control group (baseline)
    control_sent <- sample(
      rnorm(n, mean = mean_control, sd = 3),
      size = n,
      replace = TRUE
    )

    # Probability of LLM account increases with sent amount (e.g., logistic or linear scaling)
    control_llm_prob <- scales::rescale(control_sent, to = c(0.2, 0.7)) # adjust range as needed
    control_llm_account <- rbinom(n, 1, prob = control_llm_prob)

    # Sample treatment group (LLM treatment)
    treatment_sent <- sample(
      rnorm(n, mean = mean_treatment, sd = 3),
      size = n,
      replace = TRUE
    )

    treatment_llm_prob <- scales::rescale(treatment_sent, to = c(0.4, 0.95)) # higher base probability
    treatment_llm_account <- rbinom(n, 1, prob = treatment_llm_prob)

    # Combine into a single data frame
    sim_data <- data.frame(
      group = rep(c("control", "treatment"), each = n),
      sent = c(control_sent, treatment_sent),
      llm_account = c(control_llm_account, treatment_llm_account)
    )

    # Fit linear regression: sent ~ group + llm_account + group:llm_account
    sim_data$group <- factor(sim_data$group, levels = c("control", "treatment"))
    fit <- lm(sent ~ group + llm_account + group:llm_account, data = sim_data)
    reg_summary <- summary(fit)$coefficients

    # Extract p-values for group_treatment, llm_account, and interaction
    reg_pvalues_int$group_treatment[i] <- if ("grouptreatment" %in% rownames(reg_summary)) reg_summary[
    reg_pvalues_int$llm_account[i] <- if ("llm_account" %in% rownames(reg_summary)) reg_summary["llm_acc
    reg_pvalues_int$interaction[i] <- if ("grouptreatment:llm_account" %in% rownames(reg_summary)) reg_

    # Wilcoxon test between control and treatment groups, separately for each llm_account value
    for (llm_val in c(0, 1)) {
      subset_data <- sim_data[sim_data$llm_account == llm_val, ]
      if (length(unique(subset_data$group)) == 2) {
        wilcox_res <- wilcox.exact(sent ~ group, data = subset_data)
        wilcox_pvalues_by_llm[i, paste0("llm", llm_val)] <- wilcox_res$p.value
      }
    }
    pb$tick()
  }

  saveRDS(reg_pvalues_int, reg_pvalues_int_file)
  saveRDS(wilcox_pvalues_by_llm, wilcox_pvalues_by_llm_file)
}

# Power of the tests
power_results_int <- data.frame(
  Test = c(
    "t-test (group)",
    "t-test (interaction)",
    "Wilcoxon Exact (llm_account=0)",
    "Wilcoxon Exact (llm_account=1)"
  ),
  Power = c(
    round(mean(reg_pvalues_int$group_treatment < 0.05, na.rm = TRUE), 3),
    round(mean(reg_pvalues_int$interaction < 0.05, na.rm = TRUE), 3),
    round(mean(wilcox_pvalues_by_llm$llm0 < 0.05, na.rm = TRUE), 3),
    round(mean(wilcox_pvalues_by_llm$llm1 < 0.05, na.rm = TRUE), 3)
  )
)

print(power_results_int)
```

```
##                             Test Power
## 1                 t-test (group) 0.612
## 2           t-test (interaction) 0.059
## 3 Wilcoxon Exact (llm_account=0) 0.597
## 4 Wilcoxon Exact (llm_account=1) 0.826
```

Simulating Punishment Decisions

```r
# SIMULATION 7: Punishment Decisions - Treatment vs Control

set.seed(123) # for reproducibility
n <- 100
reps <- 1000
mean_treatment <- 0
mean_control <- 1.5
sd_punishment <- 0.5

# File paths for cached results
reg_pvalues_file <- "reg_pvalues_sim7.rds"
wilcox_pvalues_file <- "wilcox_pvalues_sim7.rds"
last_sim_data_file <- "last_sim_data_sim7.rds"

if (file.exists(reg_pvalues_file) && file.exists(wilcox_pvalues_file) && file.exists(last_sim_data_file)
  reg_pvalues <- readRDS(reg_pvalues_file)
  wilcox_pvalues <- readRDS(wilcox_pvalues_file)
  last_sim_data <- readRDS(last_sim_data_file)
} else {
  reg_pvalues <- numeric(reps)
  wilcox_pvalues <- numeric(reps)
  last_sim_data <- NULL

  pb <- progress_bar$new(
    format = "  Simulating [:bar] :percent eta: :eta",
    total = reps, clear = FALSE, width = 60
  )

  for (i in 1:reps) {
    punishment_treatment <- rnorm(n, mean = mean_treatment, sd = sd_punishment)
    punishment_control <- rnorm(n, mean = mean_control, sd = sd_punishment)
    group <- factor(rep(c("treatment", "control"), each = n), levels = c("control", "treatment"))
    punishment <- c(punishment_treatment, punishment_control)
    sim_data <- data.frame(
      group = group,
      punishment = punishment
    )

    # Linear regression: punishment ~ group
    fit <- lm(punishment ~ group, data = sim_data)
    reg_summary <- summary(fit)$coefficients
    reg_pvalues[i] <- if ("grouptreatment" %in% rownames(reg_summary)) reg_summary["grouptreatment", "Pr

    # Wilcoxon exact test
    wilcox_res <- wilcox.exact(punishment ~ group, data = sim_data)
    wilcox_pvalues[i] <- wilcox_res$p.value

    # Store the last simulation data for plotting
    if (i == reps) last_sim_data <- sim_data

    pb$tick()
  }
  saveRDS(reg_pvalues, reg_pvalues_file)
  saveRDS(wilcox_pvalues, wilcox_pvalues_file)
  saveRDS(last_sim_data, last_sim_data_file)
}

# Print summary of p-values
cat("Proportion of significant results (p < 0.05):\n")
```

```
## Proportion of significant results (p < 0.05):
```

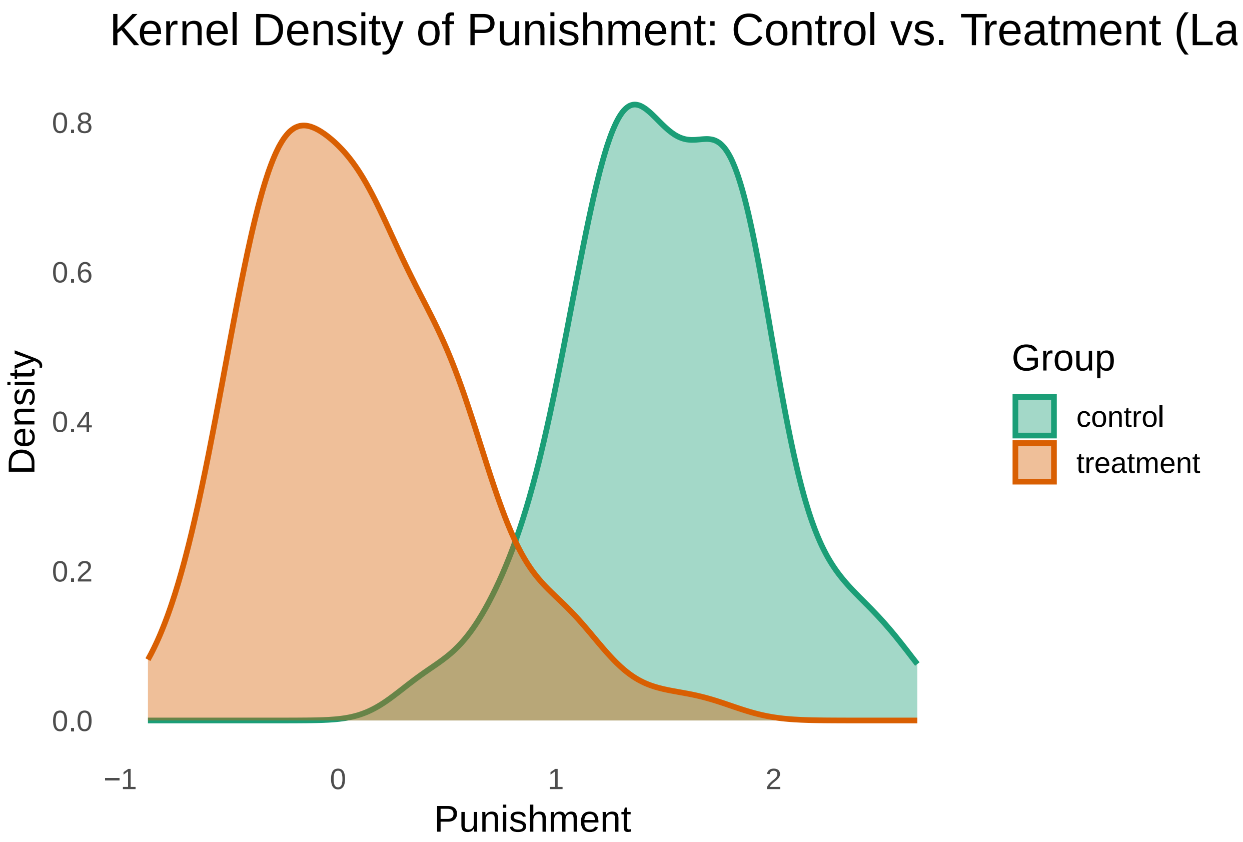
```r
cat("Linear regression:", mean(reg_pvalues < 0.05, na.rm = TRUE), "\n")
```

```
## Linear regression: 1
```

```r
cat("Wilcoxon exact:", mean(wilcox_pvalues < 0.05, na.rm = TRUE), "\n")
```

```
## Wilcoxon exact: 1
```

```r
# Plot kernel density of punishment for last simulation
library(ggplot2)
ggplot(last_sim_data, aes(x = punishment, fill = group, color = group)) +
  geom_density(alpha = 0.4, size = 1) +
  scale_fill_manual(values = c("control" = "#1b9e77", "treatment" = "#d95f02")) +
  scale_color_manual(values = c("control" = "#1b9e77", "treatment" = "#d95f02")) +
  labs(
    title = "Kernel Density of Punishment: Control vs. Treatment (Last Simulation)",
    x = "Punishment",
    y = "Density",
    fill = "Group",
    color = "Group"
  ) +
  theme_minimal(base_size = 14) +
  theme(panel.grid = element_blank())
```


Kernel Density of Punishment: Control vs. Treatment (La
0.8
0.6
0.4
0.2
0.0
Density
−1
0
1
2
Punishment
Group
control
treatment