Info 241 Preliminary Tests

2024-04-15

```
library(knitr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.0.5
```

Clean Data

```
# Read in data from each phase
phase1_treatment <- read.csv("treatment1.csv")
phase1_control <- read.csv("control1.csv")

phase2_treatment <- read.csv("treatment2.csv")
phase2_control <- read.csv("control2.csv")

phase3_treatment <- read.csv("treatment3.csv")
phase3_control <- read.csv("control3.csv")

# Add columns indicating treatment/control and phase
phase1_treatment$Phase <- 1
phase1_treatment$Group <- 1

phase1_control$Phase <- 1
phase1_control$Group <- 0

phase2_treatment$Phase <- 2
phase2_treatment$Group <- 1

phase2_control$Phase <- 2
phase2_control$Group <- 0

phase3_treatment$Phase <- 3
phase3_treatment$Group <- 1
```

```r
phase3_control$Phase <- 3
phase3_control$Group <- 0

# Combine the data frames into one
combined_data <- rbind(phase1_treatment, phase1_control, phase2_treatment, phase2_control, phase3_treatm

# modify the names
names(combined_data) = c("timestamp", "texture_a", "taste_a", "quality_a", "texture_b", "taste_b", "qual

# Optional: If you want to save the combined data to a CSV file
write.csv(combined_data, "combined_data.csv", row.names = FALSE)

# View the combined data
kable(head(combined_data))
```

timestamp	texture_a	taste_a	quality_a	texture_b	taste_b	quality_b	age	sweet_savory	hunger	vegan	dairy	exercise	phase	treatment
3/1/2024 11:50:06	9	8	8	6	6	6	21	Sweet	3	No, not vegan	Yes, I consume dairy	5-7 hours	1	1
3/1/2024 13:13:53	7	4	8	8	4	8	20	Sweet	3	No, not vegan	Yes, I consume dairy	3-5 hours	1	1
3/1/2024 13:13:58	3	2	3	5	5	5	19	Savory	1	No, not vegan	Yes, I consume dairy	0-2 hours	1	1
3/1/2024 13:14:45	10	10	10	10	10	10	21	Sweet	3	No, not vegan	Yes, I consume dairy	0-2 hours	1	1
3/1/2024 13:15:00	7	8	5	5	5	4	21	Sweet	3	No, not vegan	Yes, I consume dairy	7+ hours	1	1
3/1/2024 13:15:33	7	6	7	7	4	5	21	Savory	5	No, not vegan	Yes, I consume dairy	3-5 hours	1	1

Compare Difference in Cookie Taste/Treatment/Quality for Treatment vs. Control Across each Phase

```r
# Function to conduct t-tests on the difference in mean difference between treatment and control groups
conduct_t_tests_mean_diff <- function(phase_data, var_a, var_b) {
  treatment_mean_diff <- mean(phase_data[[var_a]] - phase_data[[var_b]])
  control_mean_diff <- mean(phase_data[[var_a]][phase_data$treatment == 0] - phase_data[[var_b]][phase_d

  t_test_result <- t.test(phase_data[[var_a]] - phase_data[[var_b]],
                          subset = phase_data$treatment == 1,
                          mu = control_mean_diff)

  return(t_test_result$p.value)
}
```

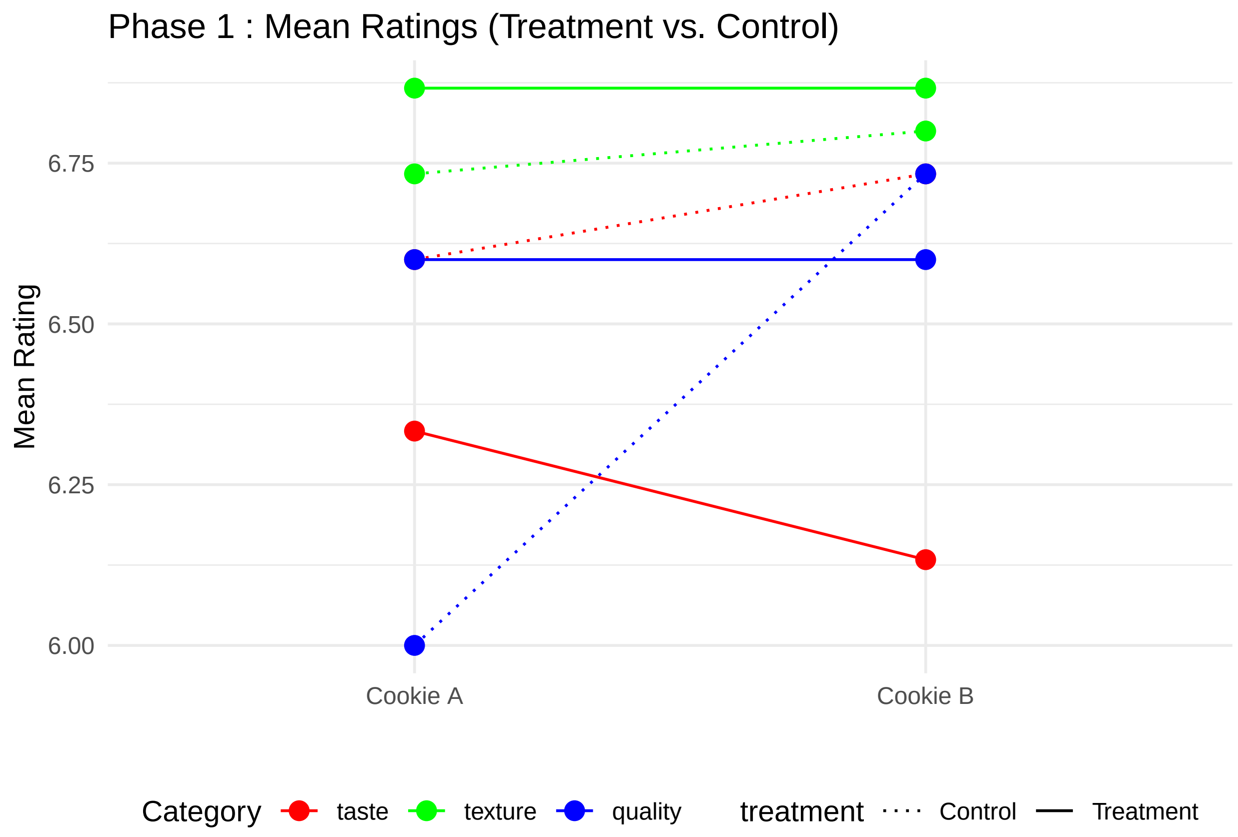
```r
# Function to generate results for each phase
generate_phase_results <- function(combined_data, vars_a, vars_b) {
  results <- data.frame(Phase = numeric(), P_value = numeric())

  for (phase in unique(combined_data$phase)) {
    phase_data <- combined_data[combined_data$phase == phase, ]
    p_values <- sapply(1:length(vars_a), function(i) conduct_t_tests_mean_diff(phase_data, vars_a[i], va

    results <- rbind(results, data.frame(Phase = phase, P_value = p_values))
  }

  return(results)
}

# Specify the variables
vars_a <- c("texture_a", "quality_a", "taste_a")
vars_b <- c("texture_b", "quality_b", "taste_b")

# Generate results for each phase
results <- round(generate_phase_results(combined_data, vars_a, vars_b),4)
results$Question <- rep(c("texture","quality","taste"),3)
results$Null <- c(rep("cookie order doesn't matter",3),rep("info doesn't matter", 3),rep("info with box

# Output the results
kable(caption="P-values for difference between treatment and control",results)
```

Table 2: P-values for difference between treatment and control

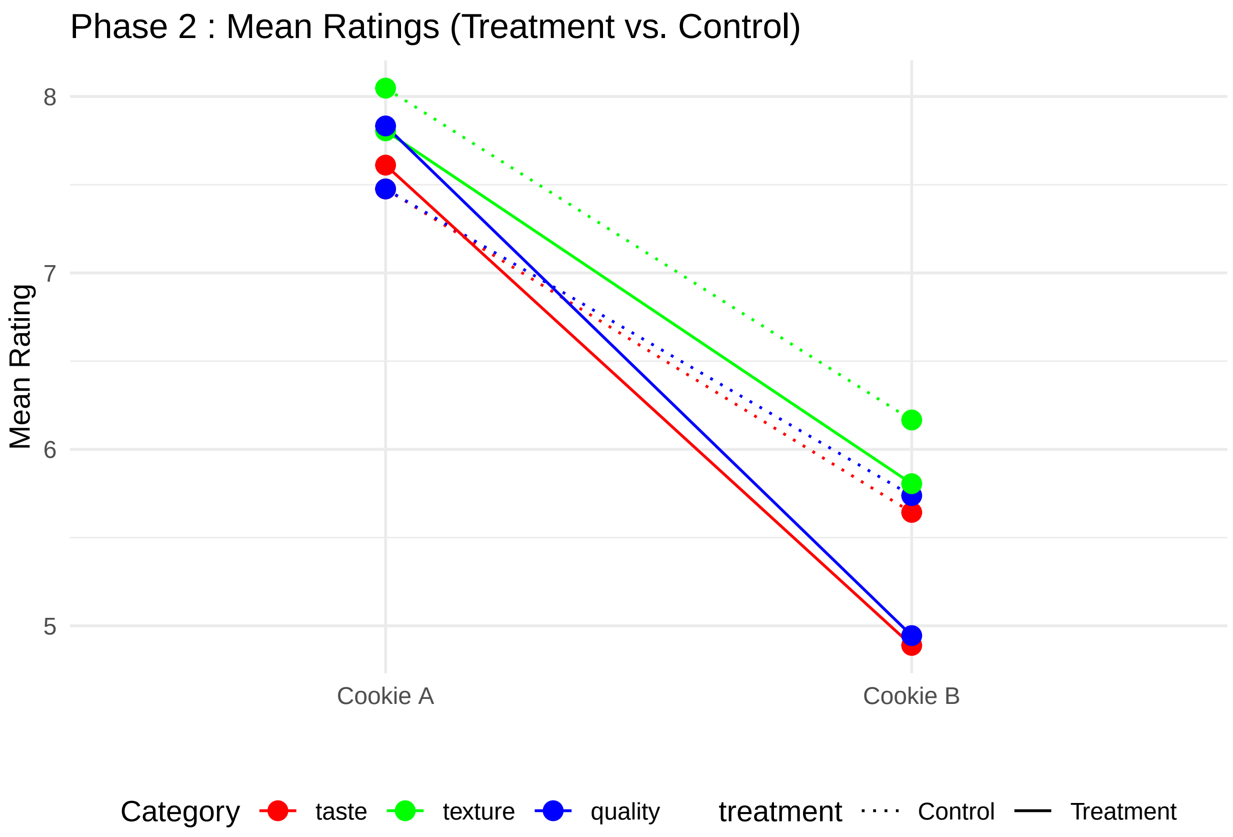
Phase	P_value	Question	Null
1	0.9465	texture	cookie order doesn’t matter
1	0.2859	quality	cookie order doesn’t matter
1	0.6901	taste	cookie order doesn’t matter
2	0.8643	texture	info doesn’t matter
2	0.0847	quality	info doesn’t matter
2	0.1840	taste	info doesn’t matter
3	0.0429	texture	info with box doesn’t matter
3	0.0004	quality	info with box doesn’t matter
3	0.0008	taste	info with box doesn’t matter

Diff in Diff Plots for Phase 1, 2, 3 Results

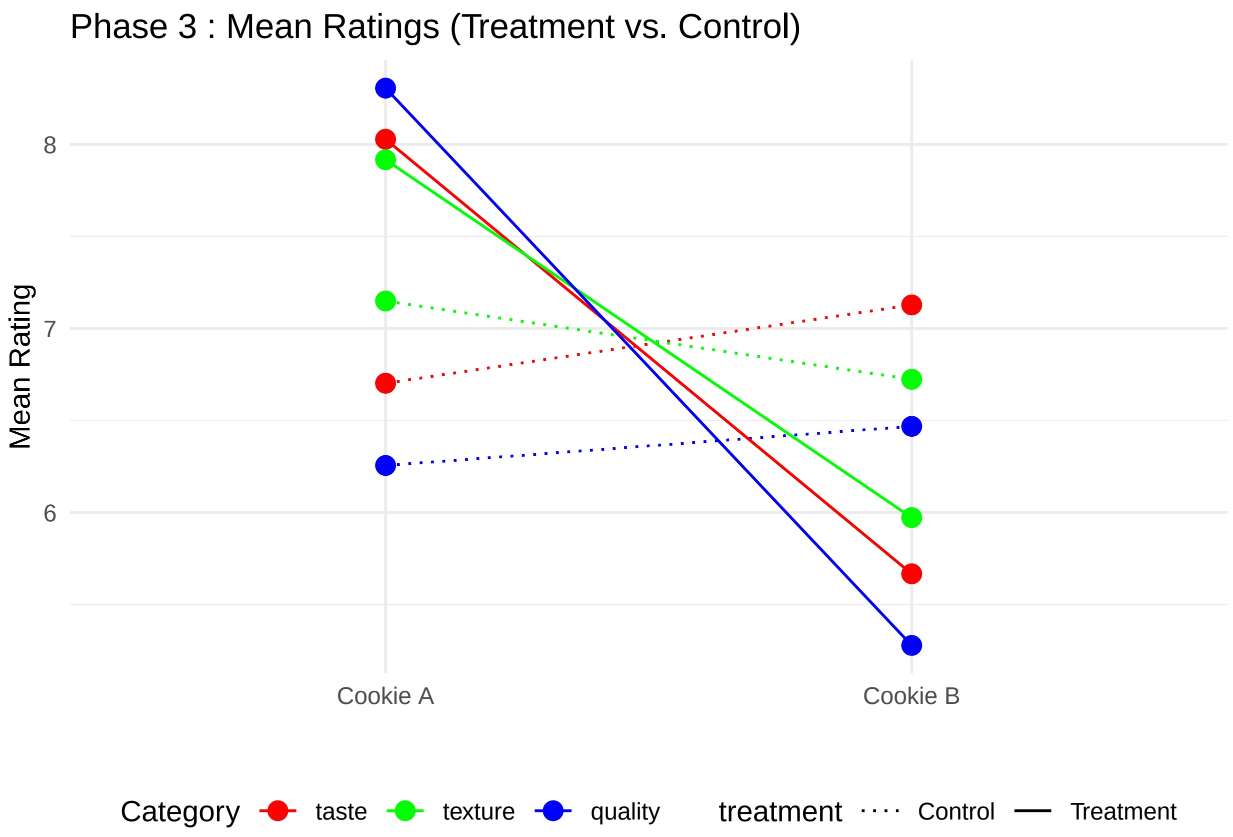
```r
# Function to generate a plot with overlaid categories for a given phase
generate_combined_categories_plot <- function(phase_data, phase_number) {
  # Filter data for the current phase
  phase_data <- phase_data[phase_data$phase == phase_number, ]

  # Calculate mean ratings for Cookie A and Cookie B for each category
  means <- phase_data %>%
    group_by(treatment) %>%
    summarise(
      taste_a = mean(taste_a),
      taste_b = mean(taste_b),
      texture_a = mean(texture_a),
      texture_b = mean(texture_b),
      quality_a = mean(quality_a),
      quality_b = mean(quality_b)
    ) %>%
    pivot_longer(cols = -treatment, names_to = c("Category", "Cookie"), names_sep = "_") %>%
    mutate(Cookie = factor(Cookie, levels = c("a", "b"), labels = c("Cookie A", "Cookie B")),
           Category = factor(Category, levels = c("taste", "texture", "quality")))

  # Convert treatment to a more descriptive factor
  means$treatment <- factor(means$treatment, levels = c(0, 1), labels = c("Control", "Treatment"))

  # Create the plot
  p <- ggplot(means, aes(x = Cookie, y = value, group = interaction(Category, treatment), color = Categ
    geom_line() +
    geom_point(size = 3) +
    scale_color_manual(values=c("red", "green", "blue")) +
    scale_linetype_manual(values=c("dotted", "solid")) +
    labs(title = paste("Phase", phase_number, ": Mean Ratings (Treatment vs. Control)"),
         x = "",
         y = "Mean Rating") +
    theme_minimal() +
    theme(legend.position = "bottom")

  return(p)
}

# Generate the plot
combined_categories_plot1 <- generate_combined_categories_plot(combined_data, 1)
combined_categories_plot2 <- generate_combined_categories_plot(combined_data, 2)
combined_categories_plot3 <- generate_combined_categories_plot(combined_data, 3)

# Print the plot
print(combined_categories_plot1)
```



```r
print(combined_categories_plot2)
```



```r
print(combined_categories_plot3)
```



```r
# Function to conduct t-tests on the difference in mean difference between two phases
conduct_t_tests_mean_diff_between_phases <- function(phase1_data, phase2_data) {
  phase1_mean_diff <- mean(phase1_data$taste_a - phase1_data$taste_b)
  phase2_mean_diff <- mean(phase2_data$taste_a - phase2_data$taste_b)

  t_test_result <- t.test(phase1_data$taste_a - phase1_data$taste_b,
                          phase2_data$taste_a - phase2_data$taste_b,
                          mu = phase2_mean_diff - phase1_mean_diff)

  return(t_test_result$p.value)
}

# Phase 2 Control vs Phase 3 Control
phase2_control <- combined_data[combined_data$phase == 2 & combined_data$treatment == 0, ]
phase3_control <- combined_data[combined_data$phase == 3 & combined_data$treatment == 0, ]
phase2_3_control_p_value <- conduct_t_tests_mean_diff_between_phases(phase2_control, phase3_control)

# Phase 2 Treatment vs Phase 3 Treatment
phase2_treatment <- combined_data[combined_data$phase == 2 & combined_data$treatment == 1, ]
phase3_treatment <- combined_data[combined_data$phase == 3 & combined_data$treatment == 1, ]
phase2_3_treatment_p_value <- conduct_t_tests_mean_diff_between_phases(phase2_treatment, phase3_treatmen

# Output the results
results <- data.frame(
  Null = c("Box with no info doesn't matter", "Box with info doesn't matter"),
  P_value = round(c(phase2_3_control_p_value, phase2_3_treatment_p_value),4)
)

kable(caption= "P-values for difference between Phase 2 and 3 Results",results)
```

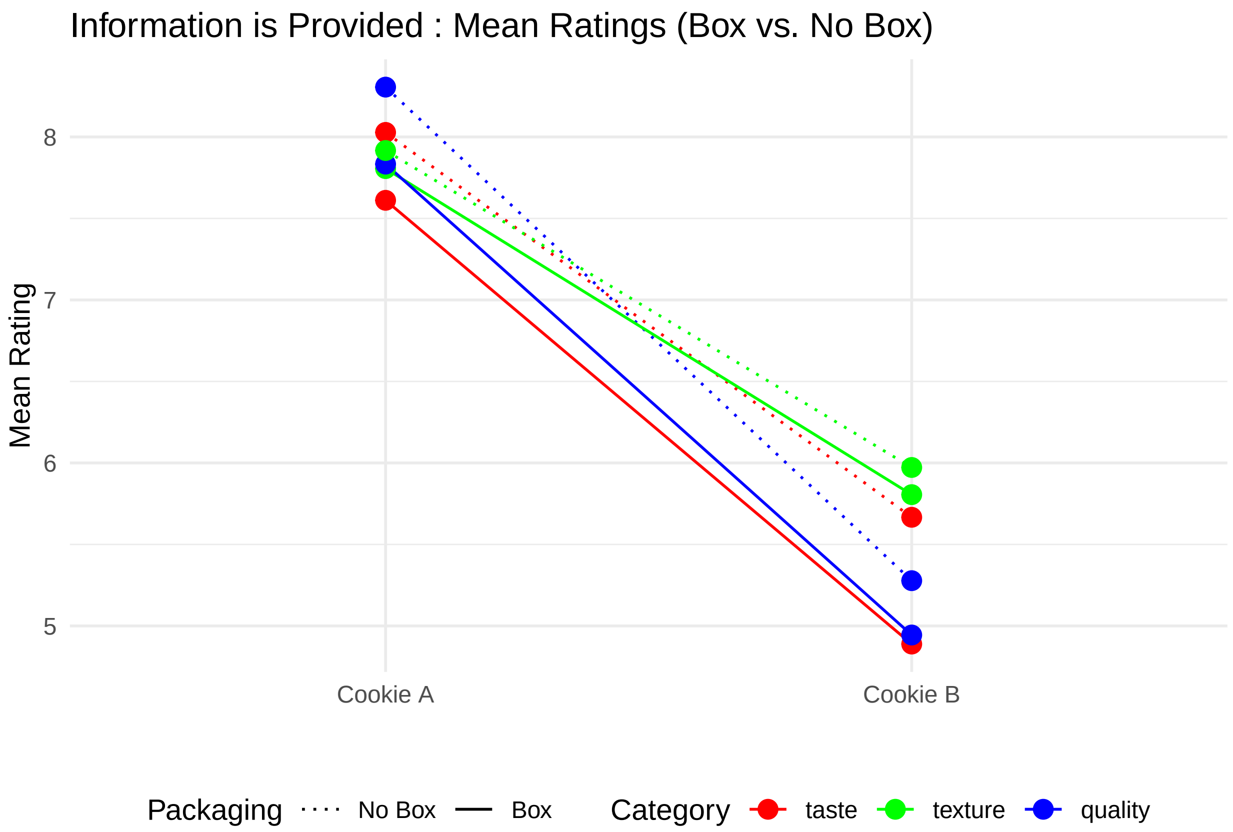
Table 3: P-values for difference between Phase 2 and 3 Results

Null	P_value
Box with no info doesn't matter	0.0000
Box with info doesn't matter	0.2583

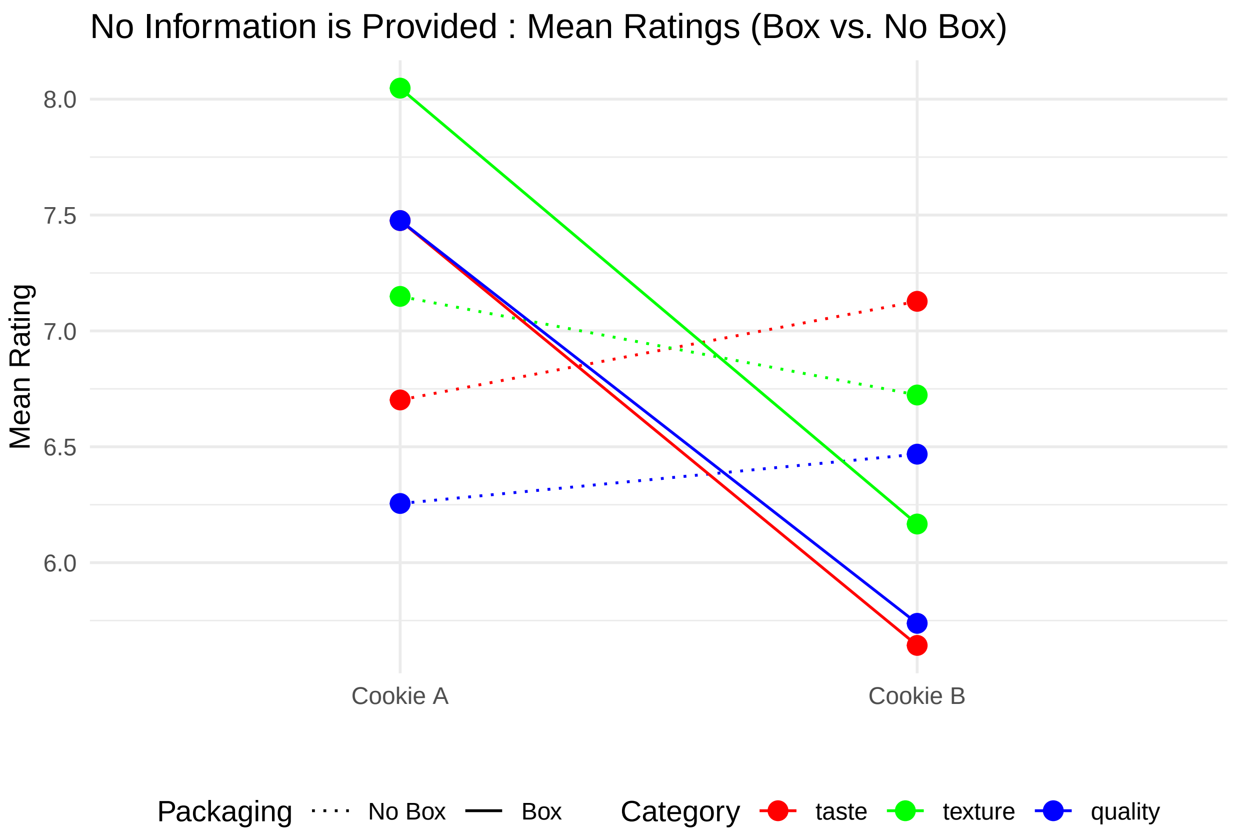
Diff in Diff Plots Comparing Across Phase 3 and Phase 2

```
# Function to generate a plot with overlaid categories for a given phase
generate_combined_categories_plot <- function(phase_data, treat) {
  # Filter data for the current phase
  phase_data <- phase_data[phase_data$treatment == treat, ]

  # Calculate mean ratings for Cookie A and Cookie B for each category
  means <- phase_data %>%
    group_by(phase) %>%
    summarise(
      taste_a = mean(taste_a),
      taste_b = mean(taste_b),
      texture_a = mean(texture_a),
      texture_b = mean(texture_b),
      quality_a = mean(quality_a),
      quality_b = mean(quality_b)
    ) %>%
    pivot_longer(cols = -phase, names_to = c("Category", "Cookie"), names_sep = "_") %>%
    mutate(Cookie = factor(Cookie, levels = c("a", "b"), labels = c("Cookie A", "Cookie B")),
           Category = factor(Category, levels = c("taste", "texture", "quality")))

  # Convert treatment to a more descriptive factor
  means$Packaging <- factor(means$phase, levels = c(1, 0), labels = c("No Box", "Box"))

  # Create the plot
  title_option = c("No Information is Provided", "Information is Provided")
  p <- ggplot(means, aes(x = Cookie, y = value, group = interaction(Category, Packaging), color = Catego
    geom_line() +
    geom_point(size = 3) +
    scale_color_manual(values=c("red", "green", "blue")) +
    scale_linetype_manual(values=c("dotted", "solid")) +
    labs(title = paste(title_option[treat+1], ": Mean Ratings (Box vs. No Box)"),
         x = "",
         y = "Mean Rating") +
    theme_minimal() +
    theme(legend.position = "bottom")

  return(p)
}

# Generate the plot
```

```
data23 = filter(combined_data, phase>1)
data23$phase = data23$phase-2
combined_categories_plot1 <- generate_combined_categories_plot(data23, 1)
combined_categories_plot2 <- generate_combined_categories_plot(data23, 0)
print(combined_categories_plot1)
```



```
print(combined_categories_plot2)
```



No Information is Provided : Mean Ratings (Box vs. No Box)