Cookies_Analysis_Coding-Sample.R

gabrielsolis

2024-11-16

```
################################################################################
###
# Cookie Privacy Research Analysis
#
# Author: Gabriel Solis (solisgab@usc.edu)
# Created: 2023-07-18
#
# Description:
#   This script analyzes survey data on consumer behavior regarding cookie
consent
#   and online privacy. It investigates how firm size affects user trust and
#   cookie acceptance decisions, while controlling for privacy regulation
awareness
#   (GDPR/CCPA) and demographic factors.
#
# Dependencies:
#   - tidyverse (data manipulation and visualization)
#   - GGally (enhanced visualization)
#   - reshape2 (data reshaping)
#   - cowplot (plot arrangement)
#   - infer (statistical inference)
#   - lmtest (regression diagnostics)
#
# Input Data:
#   - cookies_survey.csv: Raw survey responses
#
# Output:
#   - Statistical tests of cookie acceptance by firm size
#   - Visualizations of privacy regulation awareness
#   - Analysis of trust differences between small and large firms
################################################################################
###

# Load required packages
suppressPackageStartupMessages({
  library(tidyverse)
  library(GGally)
  library(reshape2)
  library(cowplot)
  library(infer)
  library(lmtest)
```

```r
})

################################################################################
###
# Data Import and Cleaning
################################################################################
###

# Import survey data
cookies <- read.csv("cookies_survey.csv")

# Retain rows passing two attention checks related to survey quality
cookies_clean <- cookies %>%
  filter(
    sec3_attention.check == "A large number of ‘friends’ or contacts (200+
people)" &
      Attention.Check == "Zenith Computing"
  )
# Note: Removed 52 entries that failed quality checks (from 153 to 101).


################################################################################
###
# Hypothesis Testing: Cookie Acceptance by Firm Size
################################################################################
###

# Prepare data for firm size comparison
small_large_table <- cookies_clean %>%
  filter(stimulus.id %in% c("small-old-ads", "large-old-ads")) %>%
  mutate(stimulus.id = factor(stimulus.id,
                              levels = c("large-old-ads", "small-old-ads")))
%>%
  select(stimulus.id, DV_accept.3)

# Calculate observed difference in means
obs_stat <- small_large_table %>%
  specify(DV_accept.3 ~ stimulus.id) %>%
  calculate(stat = "diff in means",
            order = c("large-old-ads", "small-old-ads"))

# Generate null distribution (5000 permutations)
null_dist <- small_large_table %>%
  specify(DV_accept.3 ~ stimulus.id) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 5000) %>%
  calculate(stat = "diff in means",
            order = c("large-old-ads", "small-old-ads"))
```

```
## Setting `type = "permute"` in `generate()`.

# Calculate and display p-value
p_value <- null_dist %>%
  get_p_value(obs_stat = obs_stat, direction = "two-sided")

##############################################################################
###
# Visualization: Privacy Regulation Awareness
##############################################################################
###

# Prepare data frame for GDPR and CCPA knowledge comparison
GDPR_CCPA_knowledge <- data.frame(
  Regulation = c(rep("GDPR", length(cookies_clean$GDPR.Knowledge)),
                 rep("CCPA", length(cookies_clean$CCPA.Knowledge))),
  Response = c(cookies_clean$GDPR.Knowledge,
               cookies_clean$CCPA.Knowledge)
) %>%
  count(Regulation, Response, name = "Total")

# Create visualization
regulation_plot <- GDPR_CCPA_knowledge %>%
  ggplot(aes(x = reorder(Response, -Total / 101),
             y = Total / 101,
             fill = Regulation)) +
  geom_bar(color = "white", position = "dodge", stat = "identity") +
  labs(
    x = "Knowledge Level",
    y = "Proportion of Responses",
    title = "Knowledge of Privacy Regulations",
    subtitle = "Comparison of GDPR and CCPA Awareness"
  ) +
  scale_fill_manual(values = c("#bd0026", "#0868ac")) +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
  theme_minimal(base_size = 12) +
  theme(
    axis.text.x = element_text(angle = 30, hjust = 1),
    plot.title = element_text(face = "bold"),
    plot.subtitle = element_text(color = "gray50")
  )

##############################################################################
###
# Analysis: Trust in Large vs Small Firms
##############################################################################
###

# Calculate trust scores and standard errors
trust_scores <- data.frame(
```

```r
  Company = c("Amazon", "Google", "Apple", "Meta", "TikTok"),
  Mean = sapply(c("Amazon", "Google", "Apple", "Meta", "TikTok"),
                function(x) mean(cookies_clean[[x]], na.rm = TRUE)),
  SE = sapply(c("Amazon", "Google", "Apple", "Meta", "TikTok"),
              function(x) sd(cookies_clean[[x]], na.rm = TRUE) /
                sqrt(sum(!is.na(cookies_clean[[x]]))))
)

# Create visualization
trust_plot <- trust_scores %>%
  ggplot(aes(x = reorder(Company, -Mean), y = Mean, fill = Company)) +
  geom_bar(stat = "identity", color = "white") +
  geom_errorbar(aes(ymin = Mean - SE, ymax = Mean + SE), width = 0.2) +
  labs(
    x = NULL,
    y = "Average Trust Score",
    title = "Consumer Trust Across Tech Companies",
    subtitle = "Error bars represent ±1 standard error"
  ) +
  theme_minimal(base_size = 12) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "none",
    plot.title = element_text(face = "bold")
  )

################################################################################
###
# Save processed data and results
################################################################################
###

# Save cleaned dataset
write.csv(cookies_clean, "cookies_clean.csv", row.names = FALSE)

# Save plots
ggsave("regulation_awareness.pdf", regulation_plot, width = 10, height = 6)
ggsave("trust_scores.pdf", trust_plot, width = 10, height = 6)

# Print session info for reproducibility
################################################################################
###
# Composite Score Analysis
################################################################################
###

# Create composite dependent variable
cookies_clean <- cookies_clean %>%
  mutate(
```

```
    # Average of comfort-related measures, handling NAs appropriately
    Composite_DV = rowMeans(select(., starts_with("DV_comfort")), na.rm =
TRUE)
  )

# Hypothesis test for composite scores
composite_test <- cookies_clean %>%
  filter(stimulus.id %in% c("small-old-ads", "large-old-ads")) %>%
  {
    # Capture observed statistic
    obs_stat <- specify(., Composite_DV ~ stimulus.id) %>%
      calculate(stat = "diff in means",
                order = c("large-old-ads", "small-old-ads"))

    # Generate null distribution
    null_dist <- specify(., Composite_DV ~ stimulus.id) %>%
      hypothesize(null = "independence") %>%
      generate(reps = 5000) %>%
      calculate(stat = "diff in means",
                order = c("large-old-ads", "small-old-ads"))

    # Return results as list
    list(
      observed = obs_stat,
      null_distribution = null_dist,
      p_value = get_p_value(null_dist, obs_stat, direction = "two-sided")
    )
  }

## Setting `type = "permute"` in `generate()`.

##############################################################################
###
# Cookie Knowledge Analysis
##############################################################################
###

# Analyze True/False responses about cookie knowledge
knowledge_analysis <- list(
  q1 = sum(cookies_clean$sec2_T.F.cookies == "False", na.rm = TRUE),
  q2 = sum(cookies_clean$sec2_T.F.cookies_2 == "True", na.rm = TRUE),
  q3 = sum(cookies_clean$sec2_T.F.cookies.3 == "False", na.rm = TRUE),
  q4 = sum(cookies_clean$sec2_T.F.cookies_4 == "True", na.rm = TRUE),
  q5 = sum(cookies_clean$sec2_T.F.cookies_5 == "False", na.rm = TRUE),
  q6 = sum(cookies_clean$sec2_T.F.cookies_6 == "True", na.rm = TRUE)
)

# Calculate percentage of correct responses
knowledge_summary <- data.frame(
  Question = paste0("Q", 1:6),
```

```r
  Incorrect = unlist(knowledge_analysis),
  Total = nrow(cookies_clean),
  Percent_Correct = 100 * (1 - unlist(knowledge_analysis) /
nrow(cookies_clean))
)

##############################################################################
###
# Trust Analysis by Firm Type
##############################################################################
###

# Prepare data for trust analysis
trust_analysis <- data.frame(
  Firm_Type = rep(c("Small", "Large"), each = nrow(cookies_clean)),
  Trust_Score = c(cookies_clean$Start_up,
cookies_clean$Large_Established_Firm)
) %>%
  # Remove any NA values
  filter(!is.na(Trust_Score))

# Fit linear model
trust_model <- lm(Trust_Score ~ Firm_Type, data = trust_analysis)

# Create summary statistics
trust_summary <- trust_analysis %>%
  group_by(Firm_Type) %>%
  summarise(
    Mean = mean(Trust_Score, na.rm = TRUE),
    SD = sd(Trust_Score, na.rm = TRUE),
    SE = SD / sqrt(n()),
    CI_Lower = Mean - 1.96 * SE,
    CI_Upper = Mean + 1.96 * SE,
    .groups = "drop"
  )

# Visualization of trust scores
trust_comparison_plot <- ggplot(trust_summary,
                                aes(x = Firm_Type, y = Mean, fill =
Firm_Type)) +
  geom_bar(stat = "identity", color = "black", alpha = 0.8) +
  geom_errorbar(aes(ymin = CI_Lower, ymax = CI_Upper),
                width = 0.2, color = "black") +
  labs(
    title = "Trust Scores by Firm Type",
    subtitle = "Error bars represent 95% confidence intervals",
    x = "Firm Type",
    y = "Average Trust Score"
  ) +
```

```r
  scale_fill_manual(values = c("Small" = "#bd0026", "Large" = "#0868ac")) +
  theme_minimal(base_size = 12) +
  theme(
    legend.position = "none",
    plot.title = element_text(face = "bold"),
    axis.title = element_text(face = "bold")
  )

################################################################################
###
# Demographic Analysis
################################################################################
###

# Function to create demographic summary
create_demographic_summary <- function(data, variable) {
  data %>%
    group_by(!!sym(variable)) %>%
    summarise(
      Count = n(),
      Percentage = n() / nrow(data) * 100,
      .groups = "drop"
    ) %>%
    arrange(desc(Percentage))
}

# Create summaries for each demographic variable
demographic_summaries <- list(
  Education = create_demographic_summary(cookies_clean, "Education"),
  Income = create_demographic_summary(cookies_clean, "Income"),
  Employment = create_demographic_summary(cookies_clean, "Employment"),
  Political = create_demographic_summary(cookies_clean, "Political")
)

# Age summary statistics
age_summary <- cookies_clean %>%
  summarise(
    Mean_Age = mean(as.numeric(Age), na.rm = TRUE),
    SD_Age = sd(as.numeric(Age), na.rm = TRUE),
    Median_Age = median(as.numeric(Age), na.rm = TRUE),
    Q1_Age = quantile(as.numeric(Age), 0.25, na.rm = TRUE),
    Q3_Age = quantile(as.numeric(Age), 0.75, na.rm = TRUE)
  )

################################################################################
###
# Save Results and Generate Report
################################################################################
###
```

```r
# Save all plots
ggsave("trust_comparison.pdf", trust_comparison_plot, width = 10, height = 6)

# Save summary statistics
write.csv(knowledge_summary, "cookie_knowledge_summary.csv", row.names =
FALSE)
write.csv(trust_summary, "trust_analysis_summary.csv", row.names = FALSE)

# Save demographic summaries
lapply(names(demographic_summaries), function(name) {
  write.csv(demographic_summaries[[name]],
            paste0("demographic_", tolower(name), ".csv"),
            row.names = FALSE)
})
```

```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
##
## [[4]]
## NULL
```

```r
# Print analysis summaries
cat("\n\nTrust Model Summary:\n")
```

```
##
##
## Trust Model Summary:
```

```r
print(summary(trust_model))
```

```
##
## Call:
## lm(formula = Trust_Score ~ Firm_Type, data = trust_analysis)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0297 -1.0297  0.1089  1.1089  4.1089
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      4.0297     0.1696  23.767  < 2e-16 ***
## Firm_TypeSmall  -1.1386     0.2398  -4.748  3.9e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 1.704 on 200 degrees of freedom
## Multiple R-squared:  0.1013, Adjusted R-squared:  0.09683
## F-statistic: 22.55 on 1 and 200 DF,  p-value: 3.902e-06

cat("\n\nDemographic Summary:\n")

##
##
## Demographic Summary:

print(age_summary)

##   Mean_Age   SD_Age Median_Age Q1_Age Q3_Age
## 1 41.70297 15.09937         36     29     53

# Print session info for reproducibility
################################################################################
###
# Privacy Notice and Engagement Analysis
################################################################################
###

# Analyze privacy notice responses and engagement time
privacy_analysis <- tibble(
  Notice = cookies_clean$sec2_privacy.notice,
  Time = cookies_clean$sec2_cookie.time
) %>%
  count(Notice, Time, name = "Total") %>%
  mutate(
    Notice = reorder(Notice, -Total),
    Time = factor(Time, levels = c(
      "Less than 10 seconds",
      "10-30 seconds",
      "20-30 seconds",
      "30 seconds to 1 minute",
      "1-2 minutes",
      "More than 2 minutes",
      "Not applicable"
    ))
  )

# Create visualization for privacy notice engagement
privacy_engagement_plot <- privacy_analysis %>%
  ggplot(aes(x = Notice, y = Total, fill = Time)) +
  geom_bar(color = "white", position = "dodge", stat = "identity") +
  labs(
    title = "Privacy Notice Engagement Patterns",
    subtitle = "Analysis of user interaction time with privacy notices",
    x = "Privacy Notice Type",
    y = "Number of Responses",
```

```r
    fill = "Time Spent"
  ) +
  scale_fill_brewer(palette = "RdYlBu") +
  theme_minimal(base_size = 12) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    plot.title = element_text(face = "bold"),
    legend.position = "right"
  )

################################################################################
###
# Tech Company Trust Analysis
################################################################################
###

# Analyze trust levels for major tech companies
tech_trust <- data.frame(
  Company = c("Amazon", "Google", "Apple", "Meta", "TikTok"),
  Mean = c(
    mean(cookies_clean$Amazon, na.rm = TRUE),
    mean(cookies_clean$Google, na.rm = TRUE),
    mean(cookies_clean$Apple, na.rm = TRUE),
    mean(cookies_clean$Meta, na.rm = TRUE),
    mean(cookies_clean$TikTok, na.rm = TRUE)
  ),
  SE = c(
    sd(cookies_clean$Amazon, na.rm = TRUE) /
sqrt(sum(!is.na(cookies_clean$Amazon))),
    sd(cookies_clean$Google, na.rm = TRUE) /
sqrt(sum(!is.na(cookies_clean$Google))),
    sd(cookies_clean$Apple, na.rm = TRUE) /
sqrt(sum(!is.na(cookies_clean$Apple))),
    sd(cookies_clean$Meta, na.rm = TRUE) /
sqrt(sum(!is.na(cookies_clean$Meta))),
    sd(cookies_clean$TikTok, na.rm = TRUE) /
sqrt(sum(!is.na(cookies_clean$TikTok)))
  )
)

# Create visualization for tech company trust
tech_trust_plot <- tech_trust %>%
  ggplot(aes(x = reorder(Company, -Mean), y = Mean, fill = Company)) +
  geom_bar(stat = "identity", color = "white", position = "dodge") +
  geom_errorbar(aes(ymin = Mean - SE, ymax = Mean + SE), width = 0.2) +
  labs(
    title = "Trust Levels Across Major Tech Companies",
    subtitle = "Error bars represent ±1 standard error",
    x = NULL,
    y = "Average Trust Score"
```

```r
  ) +
  scale_fill_brewer(palette = "Set3") +
  theme_minimal(base_size = 12) +
  theme(
    legend.position = "none",
    axis.text.x = element_text(angle = 30, hjust = 1),
    plot.title = element_text(face = "bold")
  )

################################################################################
###
# Additional Hypothesis Testing
################################################################################
###

# Test for differences in composite privacy scores
composite_privacy_test <- cookies_clean %>%
  filter(stimulus.id %in% c("small-old-ads", "large-old-ads")) %>%
  {
    obs_stat <- specify(., Composite_DV ~ stimulus.id) %>%
      calculate(stat = "diff in means", order = c("large-old-ads", "small-
old-ads"))

    null_dist <- specify(., Composite_DV ~ stimulus.id) %>%
      hypothesize(null = "independence") %>%
      generate(reps = 5000) %>%
      calculate(stat = "diff in means", order = c("large-old-ads", "small-
old-ads"))

    list(
      observed = obs_stat,
      null_distribution = null_dist,
      p_value = get_p_value(null_dist, obs_stat, direction = "two-sided")
    )
  }

## Setting `type = "permute"` in `generate()`.

# Visualize composite privacy score distribution
composite_score_plot <- ggplot(cookies_clean, aes(x = Composite_DV)) +
  geom_histogram(binwidth = 0.5, fill = "steelblue", color = "white") +
  facet_wrap(~stimulus.id) +
  labs(
    title = "Distribution of Composite Privacy Scores",
    subtitle = "Comparison between small and large firms",
    x = "Composite Privacy Score",
    y = "Count"
  ) +
  theme_minimal(base_size = 12) +
  theme(plot.title = element_text(face = "bold"))
```

```r
################################################################################
###
# Save Final Results and Generate Report
################################################################################
###

# Save all plots
ggsave("privacy_engagement.pdf", privacy_engagement_plot, width = 12, height
= 8)
ggsave("tech_trust.pdf", tech_trust_plot, width = 10, height = 6)
ggsave("composite_scores.pdf", composite_score_plot, width = 10, height = 6)

# Save analysis results
write.csv(tech_trust, "tech_trust_analysis.csv", row.names = FALSE)
write.csv(privacy_analysis, "privacy_engagement_analysis.csv", row.names =
FALSE)

# Create summary report
cat("\nFinal Analysis Results\n")

##
## Final Analysis Results

cat("\n1. Composite Privacy Score Analysis")

##
## 1. Composite Privacy Score Analysis

cat("\n\n1. Tech Company Trust Analysis")

##
##
## 1. Tech Company Trust Analysis

print(tech_trust)

##   Company     Mean        SE
## 1  Amazon 3.841584 0.1968729
## 2  Google 3.742574 0.1973355
## 3   Apple 3.613861 0.1964891
## 4    Meta 3.158416 0.1943421
## 5  TikTok 2.762376 0.1965789

cat("\n\n2. Privacy Notice Engagement Summary")

##
##
## 2. Privacy Notice Engagement Summary

print(table(cookies_clean$sec2_cookie.time))
```

```
##
##            1-2 minutes          10-30 seconds          20-30 seconds
##                      2                     32                      3
## 30 seconds to 1 minute   Less than 10 seconds         Not applicable
##                      8                     54                      2
```

```
##############################################################################
###
# Interaction Effects Analysis
##############################################################################
###

# Analyze interaction between privacy awareness and firm size
interaction_analysis <- cookies_clean %>%
  mutate(
    GDPR_Aware = GDPR.Knowledge != "Not at all familiar",
    CCPA_Aware = CCPA.Knowledge != "Not at all familiar",
    Privacy_Aware = GDPR_Aware | CCPA_Aware
  ) %>%
  filter(stimulus.id %in% c("small-old-ads", "large-old-ads"))

# Fit interaction model
interaction_model <- lm(DV_accept.3 ~ stimulus.id * Privacy_Aware,
                        data = interaction_analysis)

# Create summary statistics for interaction effects
interaction_summary <- interaction_analysis %>%
  group_by(stimulus.id, Privacy_Aware) %>%
  summarise(
    Mean = mean(DV_accept.3, na.rm = TRUE),
    SD = sd(DV_accept.3, na.rm = TRUE),
    N = n(),
    SE = SD / sqrt(N),
    .groups = "drop"
  )

# Visualize interaction effects
interaction_plot <- ggplot(interaction_summary,
                           aes(x = stimulus.id, y = Mean, color =
Privacy_Aware, group = Privacy_Aware)) +
  geom_point(size = 3) +
  geom_line() +
  geom_errorbar(aes(ymin = Mean - SE, ymax = Mean + SE), width = 0.2) +
  labs(
    title = "Interaction Between Privacy Awareness and Firm Size",
    subtitle = "Effect on Cookie Acceptance",
    x = "Firm Size",
    y = "Average Acceptance Rate",
    color = "Privacy Aware"
  ) +
```

```r
  theme_minimal(base_size = 12) +
  theme(plot.title = element_text(face = "bold"))

################################################################################
###
# Robustness Checks
################################################################################
###

# Check for demographic effects
demographic_models <- list(
  education = lm(DV_accept.3 ~ stimulus.id + Education, data =
interaction_analysis),
  income = lm(DV_accept.3 ~ stimulus.id + Income, data =
interaction_analysis),
  age = lm(DV_accept.3 ~ stimulus.id + Age, data = interaction_analysis),
  political = lm(DV_accept.3 ~ stimulus.id + Political, data =
interaction_analysis)
)

# Create summary of robustness checks
robustness_summary <- lapply(demographic_models, function(model) {
  data.frame(
    R_squared = summary(model)$r.squared,
    F_stat = summary(model)$fstatistic[1],
    P_value = pf(
      summary(model)$fstatistic[1],
      summary(model)$fstatistic[2],
      summary(model)$fstatistic[3],
      lower.tail = FALSE
    )
  )
})

# Combine into single data frame
robustness_df <- bind_rows(robustness_summary, .id = "Model")

################################################################################
###
# Additional Comfort Scale Analysis
################################################################################
###

# Analyze comfort scales across different dimensions
comfort_analysis <- cookies_clean %>%
  select(starts_with("DV_comfort_scales_")) %>%
  gather(key = "Scale", value = "Score") %>%
  mutate(
    Scale = factor(Scale,
```

```
                    levels = paste0("DV_comfort_scales_", 1:5),
                    labels = c("Personal Info", "Location", "Browsing",
                               "Third Party", "Advertising"))
  )

# Create summary statistics for comfort scales
comfort_summary <- comfort_analysis %>%
  group_by(Scale) %>%
  summarise(
    Mean = mean(Score, na.rm = TRUE),
    SD = sd(Score, na.rm = TRUE),
    N = sum(!is.na(Score)),
    SE = SD / sqrt(N),
    .groups = "drop"
  )

# Visualize comfort scales
comfort_plot <- ggplot(comfort_summary, aes(x = reorder(Scale, -Mean), y =
Mean)) +
  geom_bar(stat = "identity", fill = "steelblue", alpha = 0.7) +
  geom_errorbar(aes(ymin = Mean - SE, ymax = Mean + SE), width = 0.2) +
  labs(
    title = "User Comfort Levels Across Different Data Types",
    subtitle = "Error bars represent ±1 standard error",
    x = "Data Type",
    y = "Average Comfort Score"
  ) +
  theme_minimal(base_size = 12) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    plot.title = element_text(face = "bold")
  )

##############################################################################
###
# Final Analysis Output
##############################################################################
###

# Create comprehensive results summary
results_summary <- list(
  "Interaction Effects" = list(
    model = summary(interaction_model),
    summary_stats = interaction_summary
  ),
  "Robustness Checks" = robustness_df,
  "Comfort Analysis" = comfort_summary
)
```

```r
# Save final results
saveRDS(results_summary, "final_analysis_results.rds")

# Save final plots
ggsave("interaction_effects.pdf", interaction_plot, width = 10, height = 6)
ggsave("comfort_analysis.pdf", comfort_plot, width = 10, height = 6)

# Create final summary report
cat("\nFinal Analysis Summary\n")
```

```
##
## Final Analysis Summary
```

```r
cat("====================\n\n")
```

```
## ====================
```

```r
cat("1. Interaction Effects Analysis\n")
```

```
## 1. Interaction Effects Analysis
```

```r
cat("   F-statistic:", summary(interaction_model)$fstatistic[1], "\n")
```

```
##    F-statistic: 1.166458
```

```r
cat("   p-value:", pf(summary(interaction_model)$fstatistic[1],
                      summary(interaction_model)$fstatistic[2],
                      summary(interaction_model)$fstatistic[3],
                      lower.tail = FALSE), "\n\n")
```

```
##    p-value: 0.284601
```

```r
cat("2. Robustness Check Summary\n")
```

```
## 2. Robustness Check Summary
```

```r
print(robustness_df)
```

```
##               Model  R_squared    F_stat    P_value
## value...1 education 0.08902472 0.8632342 0.52795187
## value...2    income 0.09533076 0.5854243 0.80270039
## value...3       age 0.09071938 2.8434593 0.06651064
## value...4 political 0.15621142 1.3752588 0.23553121
```

```r
cat("\n")

cat("3. Comfort Scale Analysis\n")
```

```
## 3. Comfort Scale Analysis
```

```r
print(comfort_summary)
```

```
## # A tibble: 5 × 5
##   Scale          Mean    SD     N    SE
```

```
##   <fct>         <dbl> <dbl> <int> <dbl>
## 1 Personal Info  4.22  1.90   101 0.189
## 2 Location       4.13  2.00   101 0.199
## 3 Browsing       4.17  1.95   101 0.194
## 4 Third Party    3.79  1.94   101 0.193
## 5 Advertising    3.94  1.99   101 0.198
```

```
# Save detailed technical appendix
sink("technical_appendix.txt")
cat("Technical Appendix\n")
```

```
## Technical Appendix
```

```
cat("=================\n\n")
```

```
## =================
```

```
cat("1. Model Specifications\n")
```

```
## 1. Model Specifications
```

```
print(summary(interaction_model))
```

```
##
## Call:
## lm(formula = DV_accept.3 ~ stimulus.id * Privacy_Aware, data =
interaction_analysis)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7000 -2.0750  0.6125  1.9250  2.9250
##
## Coefficients: (2 not defined because of singularities)
##                                            Estimate Std. Error t value
Pr(>|t|)
## (Intercept)                                  4.7000     0.4725   9.947
3.78e-14
## stimulus.idsmall-old-ads                    -0.6250     0.5787  -1.080
0.285
## Privacy_AwareTRUE                                NA         NA      NA
NA
## stimulus.idsmall-old-ads:Privacy_AwareTRUE       NA         NA      NA
NA
##
## (Intercept)                                ***
## stimulus.idsmall-old-ads
## Privacy_AwareTRUE
## stimulus.idsmall-old-ads:Privacy_AwareTRUE
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.113 on 58 degrees of freedom
```

```
## Multiple R-squared:  0.01971,    Adjusted R-squared:  0.002813
## F-statistic: 1.166 on 1 and 58 DF,  p-value: 0.2846
```

```
cat("\n2. Robustness Checks\n")
```

```
##
## 2. Robustness Checks
```

```
lapply(demographic_models, summary)
```

```
## $education
##
## Call:
## lm(formula = DV_accept.3 ~ stimulus.id + Education, data =
interaction_analysis)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8625 -2.1906  0.2273  1.7040  3.8094
##
## Coefficients:
##                                         Estimate Std. Error t value
Pr(>|t|)
## (Intercept)                              4.86245    0.53477   9.093
2.1e-12
## stimulus.idsmall-old-ads                -0.63429    0.61575  -1.030
0.308
## EducationHigh School                     0.09041    0.68411   0.132
0.895
## EducationMaster's Degree                -1.03751    0.76432  -1.357
0.180
## EducationPh.D., M.D., or other doctorate  1.45469    1.56030   0.932
0.355
## EducationSome High School               -0.22816    1.58184  -0.144
0.886
## EducationTrade School                   -1.86245    2.19700  -0.848
0.400
##
## (Intercept)                             ***
## stimulus.idsmall-old-ads
## EducationHigh School
## EducationMaster's Degree
## EducationPh.D., M.D., or other doctorate
## EducationSome High School
## EducationTrade School
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.131 on 53 degrees of freedom
## Multiple R-squared:  0.08902,    Adjusted R-squared:  -0.0141
## F-statistic: 0.8632 on 6 and 53 DF,  p-value: 0.528
```

```
##
##
## $income
##
## Call:
## lm(formula = DV_accept.3 ~ stimulus.id + Income, data =
interaction_analysis)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6808 -1.8868  0.6144  1.5690  3.1132
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 4.2948     0.8347   5.145 4.49e-06 ***
## stimulus.idsmall-old-ads   -0.5896     0.6299  -0.936    0.354
## Income$100,000-$124,999     1.4707     1.1324   1.299    0.200
## Income$125,000-$149,999    -1.0000     1.7284  -0.579    0.565
## Income$150,000-$199,999     1.7052     2.3402   0.729    0.470
## Income$200,000 and up       2.2948     2.3402   0.981    0.332
## Income$25,000-$49,999       0.2977     0.9395   0.317    0.753
## Income$50,000-$74,999       0.1816     1.0034   0.181    0.857
## Income$75,000-$99,999       0.3860     1.0767   0.359    0.721
## IncomePrefer not to say    -0.2052     1.7569  -0.117    0.907
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.186 on 50 degrees of freedom
## Multiple R-squared:  0.09533,    Adjusted R-squared:  -0.06751
## F-statistic: 0.5854 on 9 and 50 DF,  p-value: 0.8027
##
##
## $age
##
## Call:
## lm(formula = DV_accept.3 ~ stimulus.id + Age, data = interaction_analysis)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3371 -1.6155  0.5243  1.6255  3.6390
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                6.24199    0.86308   7.232 1.31e-09 ***
## stimulus.idsmall-old-ads -0.67296    0.56266  -1.196   0.2366
## Age                      -0.03620    0.01716  -2.110   0.0393 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.053 on 57 degrees of freedom
```

```
## Multiple R-squared:  0.09072,    Adjusted R-squared:  0.05881
## F-statistic: 2.843 on 2 and 57 DF,  p-value: 0.06651
##
##
## $political
##
## Call:
## lm(formula = DV_accept.3 ~ stimulus.id + Political, data =
interaction_analysis)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5807 -1.7060  0.4193  1.1543  4.0914
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      3.8821     0.8158   4.759 1.59e-05 ***
## stimulus.idsmall-old-ads        -0.4114     0.5761  -0.714    0.478
## PoliticalLiberal                 1.3921     1.0064   1.383    0.172
## PoliticalNo preference           0.8403     1.0728   0.783    0.437
## PoliticalSlightly conservative   1.3750     1.0352   1.328    0.190
## PoliticalSlightly liberal        0.2353     0.9639   0.244    0.808
## PoliticalVery conservative       1.6986     1.0377   1.637    0.108
## PoliticalVery liberal           -0.5622     1.0099  -0.557    0.580
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.07 on 52 degrees of freedom
## Multiple R-squared:  0.1562, Adjusted R-squared:  0.04262
## F-statistic: 1.375 on 7 and 52 DF,  p-value: 0.2355
```

```r
cat("\n3. Session Information\n")
```

```
##
## 3. Session Information
```

```r
sessionInfo()
```

```
## R version 4.4.1 (2024-06-14)
## Platform: aarch64-apple-darwin20
## Running under: macOS 15.1
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.4-
arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.4-
arm64/Resources/lib/libRlapack.dylib;  LAPACK version 3.12.0
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
```

```
## time zone: America/Los_Angeles
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lmtest_0.9-40   zoo_1.8-12      infer_1.0.7     cowplot_1.1.3
##  [5] reshape2_1.4.4  GGally_2.2.1    lubridate_1.9.3 forcats_1.0.0
##  [9] stringr_1.5.1   dplyr_1.1.4     purrr_1.0.2     readr_2.1.5
## [13] tidyr_1.3.1     tibble_3.2.1    ggplot2_3.5.1   tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##  [1] utf8_1.2.4         generics_0.1.3     stringi_1.8.4
lattice_0.22-6
##  [5] hms_1.1.3          digest_0.6.35      magrittr_2.0.3
evaluate_0.23
##  [9] grid_4.4.1         timechange_0.3.0   RColorBrewer_1.1-3
fastmap_1.2.0
## [13] plyr_1.8.9         fansi_1.0.6        scales_1.3.0
textshaping_0.4.0
## [17] cli_3.6.2          rlang_1.1.4        munsell_0.5.1      withr_3.0.0
## [21] yaml_2.3.8         tools_4.4.1        tzdb_0.4.0
colorspace_2.1-0
## [25] ggstats_0.6.0      vctrs_0.6.5        R6_2.5.1
lifecycle_1.0.4
## [29] ragg_1.3.2         pkgconfig_2.0.3    pillar_1.9.0       gtable_0.3.5
## [33] glue_1.7.0         Rcpp_1.0.12        systemfonts_1.1.0  xfun_0.44
## [37] tidyselect_1.2.1   rstudioapi_0.16.0  knitr_1.47         farver_2.1.2
## [41] htmltools_0.5.8.1  labeling_0.4.3     rmarkdown_2.27
compiler_4.4.1

sink()

##############################################################################
###
# END OF ANALYSIS
##############################################################################
###
```