Replication of Bargaining in the Shadow of Uncertainty:

Alireza Sadeghi (sadeghi.alireza@stud.hs_fresenius.de and 400793389)

Cologne,

Data Science for Business

Prof. Dr. Stephan Huber

February 2 , 2026

Author Note

Correspondence concerning this article should be addressed to Alireza Sadeghi (sadeghi.alireza@stud.hs_fresenius.de,400793389)

Replication of Bargaining in the Shadow of Uncertainty:

Abstract

This report presents a replication study of the dataset “Bargaining in the Shadow of Uncertainty.” The objective of the analysis was to reproduce and validate the original findings by applying the same data preparation and analytical procedures described in the underlying study. The dataset was examined and preprocessed to ensure consistency with the original methodology, including the handling of missing values and variable transformations. Exploratory data analysis and statistical techniques were used to replicate key results related to bargaining behavior under conditions of uncertainty. The replicated outcomes largely align with the original findings, supporting the robustness of the reported relationships. This replication contributes to transparency and reliability in empirical research by confirming that the original results can be reproduced using the provided data and methods.

Table of Contents

Introduction

This report was developed using RStudio, which served as the main environment for data analysis, visualization, and documentation. RStudio was chosen due to its integration of scripting, output visualization, and reproducible research tools, making it well suited for both(Agranov et al., 2024) data analysis and learning oriented reporting.(Posit Team, 2023)

The analysis began with importing the Bargaining in the Shadow of Uncertainty dataset into R(R Core Team, 2024a). Initial steps focused on familiarization with the data structure using basic inspection functions to examine variable types, summary statistics, and potential data quality issues. This stage was essential for understanding the content of the dataset and aligning the analysis with the methodology of the original study.

Subsequently, data preprocessing was performed to prepare the dataset for replication. This included handling missing values, renaming variables where necessary for clarity, and transforming variables to match the specifications used in the original analysis. Throughout this process, R scripts were used to ensure transparency and reproducibility of each step(Wickham et al., 2019).

Exploratory data analysis was then conducted using descriptive statistics and visualizations generated in R. These tools helped identify patterns and relationships relevant to bargaining behavior under uncertainty and allowed for an initial comparison with the results reported in the original study. Statistical analyses were implemented using R's built-in and external packages to replicate the key findings as closely as possible(Wickham & Grolemund, 2017).

Finally, the results and interpretations were compiled into this report, allowing the entire analytical process from data import to final results to be clearly documented and reproducible. By using RStudio, the report not only presents the replicated outcomes but also provides insight into the analytical workflow, making it accessible to readers who wish to understand or repeat the analysis themselves.(Posit Team, 2023)

Main Body

Use of RStudio and R Packages

All analyses in this project were conducted using RStudio, which provided an integrated environment for writing code, executing analyses, and documenting results. RStudio was particularly useful for this learning-oriented replication report, as it allowed the combination of narrative text, code, and visual outputs within a single reproducible workflow.

Several R packages were used to support different stages of the analysis. The readxl package was employed to import data files provided in Excel format. This package enabled efficient and reliable data loading while preserving variable structures, which is essential when working with externally provided datasets(Wickham et al., 2019). Learning how to import and inspect raw data using readxl represents an important practical skill for real-world data analysis projects.

For data manipulation and preparation, packages from the tidyverse ecosystem were used. In particular, dplyr was applied to filter observations, select relevant variables, and create new variables required for replication. These functions helped structure the dataset in a clear and readable way, making the analytical steps easier to understand and reproduce(Wickham et al., 2019). Where necessary, data reshaping techniques were applied to align the dataset with the structure described in the original study(Agranov et al., 2024).

Data visualization was primarily carried out using the ggplot2 package. This package was used to create a variety of plots, including bar charts, stacked bar charts, and comparative visualizations across different bargaining rules and session types. The layered grammar of graphics provided by ggplot2 allowed the construction of clear and informative plots that supported the interpretation of bargaining behavior under uncertainty(Wickham et al., 2019). Visualizations played a key role in verifying whether the replicated patterns were consistent with those reported in the original study(Agranov et al., 2024).

Additional supporting packages were used to facilitate workflow organization and reporting. These packages helped ensure that the analysis was transparent, reproducible, and clearly documented. By relying on widely used and well-documented R packages, the analysis followed best practices in data science and empirical research(R Core Team, 2024a) (Wickham et al., 2019).

Learning-Oriented Analytical Process

The analytical process followed a structured workflow that emphasized learning and reproducibility. After reviewing the README documentation accompanying the dataset, the data were imported into R and inspected to gain an initial understanding of their structure and content. This step was crucial for aligning the analysis with the methodology of the original study(Agranov et al., 2024).

Subsequently, data cleaning and preparation were performed using R scripts. Rather than making manual changes, all transformations were executed through code, ensuring that the entire process could be reproduced by rerunning the script. Throughout this stage, the use of tidyverse functions improved code readability and supported a step-by-step understanding of the workflow(Wickham et al., 2019).

Exploratory data analysis was then conducted using descriptive statistics and graphical methods. The plots created with ggplot2 helped identify patterns across bargaining institutions and served as a visual check of the replication results. By reproducing tables and figures similar to those presented in the original study, the analysis demonstrated consistency with the reported findings(Wickham & Grolemund, 2017) (Agranov et al., 2024).

Overall, the use of RStudio and multiple R packages allowed the analysis to be conducted in a systematic and transparent manner. This approach not only supported the replication of empirical results but also enhanced the learning experience by making each analytical step explicit and understandable to the reader(R Core Team, 2024a) (Posit Team, 2023).

Process

Understanding the Original Workflow

After reviewing the README documentation, it became clear that the data construction process relied primarily on two Stata scripts:

prepare_bargaining.do

prepare_chats.do

These scripts represent the core of the original data preparation workflow. Their purpose is to transform multiple raw experimental files into a single, clean dataset that can be used for statistical analysis. Understanding the role of each script was an essential first step before attempting to translate the code into R.

```
# Load the readxl package (install if needed)
install.packages("readxl")  # Run once if not installed
library(readxl)

# Import the Excel file, using the first row as column names
data <- read_excel("E:/My education/UNI/edu uni/Semester 2/Data science for Business/dat


# Load necessary package
install.packages("dplyr")
library(dplyr)

# Assuming your dataset is called 'data'
data <- data %>%
  # Create 'id' as a grouped identifier like egen group()
  mutate(id = as.numeric(factor(paste(sessioncode, memberid, sep = "_"))),

         # Generate treatment indicators
         m24 = (votingtreatment == "majority" & pietreatment == 24),
         m48 = (votingtreatment == "majority" & pietreatment == 48),
         m96 = (votingtreatment == "majority" & pietreatment == 96),

         u24 = (votingtreatment == "unanimity" & pietreatment == 24),
         u48 = (votingtreatment == "unanimity" & pietreatment == 48),
         u96 = (votingtreatment == "unanimity" & pietreatment == 96))
# Load packages
library(dplyr)
library(haven)   # for saving .dta files if you need to

# Assuming your dataset is called 'data'
data <- data %>%
  # Total yes votes
  mutate(
    totalyes = vote1 + vote2 + vote3,

    # Pass condition
    pass = case_when(
      is.na(totalyes) ~ NA,
      (totalyes == 3 & votingtreatment == "unanimity") |
        (totalyes >= 2 & votingtreatment == "majority") ~ TRUE,
      TRUE ~ FALSE
    ),

    # Treatment variable
    treatment = case_when(
      m24 == 1 ~ "m24",
      m48 == 1 ~ "m48",
      m96 == 1 ~ "m96",
      u24 == 1 ~ "u24",
      u48 == 1 ~ "u48",
      u96 == 1 ~ "u96",
      TRUE ~ ""
    ),

    # Unanimity dummy
    unanimity = (votingtreatment == "unanimity"),

    # Session assignment
    session = case_when(
      sessioncode %in% c("50catrnf","7f7v4l6r","js437fzd","m1","m5","q32zfvnq") ~ 1,
      sessioncode %in% c("9gzbr7vg","ax2x1ab5","ifm6r51c","krgxtmp1","m2","m6") ~ 2,
      sessioncode %in% c("6nf5jbi5","gx0695sj","jvzb19el","m3","p66q0jhd","w2npj4ao") ~
      sessioncode %in% c("9fuzcnjg","ixhiq3bn","job0mwmb","jwlbjm39","m4","wfwdfk01") ~
      TRUE ~ NA_real_
    )
  )

# Create unique session IDs
data <- data %>%
  mutate(
    uniquesession = case_when(
      votingtreatment == "majority" & pietreatment == 24 ~ as.numeric(paste0("240", sess
      votingtreatment == "majority" & pietreatment == 48 ~ as.numeric(paste0("480", sess
      votingtreatment == "majority" & pietreatment == 96 ~ as.numeric(paste0("960", sess
      votingtreatment == "unanimity" & pietreatment == 24 ~ as.numeric(paste0("241", ses
      votingtreatment == "unanimity" & pietreatment == 48 ~ as.numeric(paste0("481", ses
      votingtreatment == "unanimity" & pietreatment == 96 ~ as.numeric(paste0("961", ses
      TRUE ~ NA_real_
    )
  )

# Save as Stata .dta file
write_dta(data, "E:/My education/UNI/edu uni/Semester 2/Data science for Business/Bargai
library(readxl)
library(dplyr)
library(haven)
library(janitor)
library(stringr)

# Define paths
base_path <- "E:/My education/UNI/edu uni/Semester 2/Data science for Business"
excel_file <- file.path(base_path, "risk.xlsx")
output_file <- file.path(base_path, "risk.dta")

# Function to import one treatment group (e.g., U1_s1-U1_s4)
import_group <- function(prefix, treatment, unanimity, pie = NA, n_sessions = 4) {
  bind_rows(lapply(1:n_sessions, function(i) {
    sheet_name <- paste0(prefix, "_s", i)
    message("Importing sheet: ", sheet_name)

    df <- read_excel(excel_file, sheet = sheet_name) %>%
      clean_names() %>%
      mutate(session = i,
             treatment = treatment,
             unanimity = unanimity,
             pie = pie)
    return(df)
  }))
}

# ---- Import all treatment groups ----
risk_data <- bind_rows(
  import_group("U1", "u24", unanimity = 1, pie = 24),
  import_group("U2", "u48", unanimity = 1, pie = 48),
  import_group("U4", "u96", unanimity = 1, pie = 96),
  import_group("M1", "m24", unanimity = 0, pie = 24),
  import_group("M2", "m48", unanimity = 0, pie = 48),
  import_group("M4", "m96", unanimity = 0, pie = 96)
)

# ---- Fix long variable names for Stata ----
names(risk_data) <- names(risk_data) %>%
  str_sub(1, 32)  # shorten to first 32 characters (Stata’s max limit)

# ---- Clean up: drop rows with missing session_code ----
if ("session_code" %in% names(risk_data)) {
  risk_data <- risk_data %>%
    filter(!(is.na(session_code) | session_code == ""))
}

# ---- Save final combined dataset ----
```

```
write_dta(risk_data, output_file)
message("  Combined risk data saved to: ", output_file)
library(haven)

# Load the Stata dataset
risk_data <- read_dta("E:/My education/UNI/edu uni/Semester 2/Data science for Business/

# View the first few rows
head(risk_data)

# Optional: open in RStudio’s data viewer
View(risk_data)
library(dplyr)
library(tidyr)
library(haven)

# Load the risk dataset
risk_data <- read_dta("E:/My education/UNI/edu uni/Semester 2/Data science for Business/

# --- Rename columns to match Stata logic ---
risk_data <- risk_data %>%
  rename(
    participantcoderisk = participant_code,        # correct column name
    memberid = participant_id_in_session,          # correct column name
    playerinvested = player_invested              # correct column name
  )
```

```r
# --- Create riskmultiplier ---
risk_data <- risk_data %>%
  mutate(
    riskmultiplier = case_when(
      subsession_investment_multiplier == 3 ~ 2,
      subsession_investment_multiplier == 2.5 ~ 1,
      TRUE ~ NA_real_
    )
  )

# --- Create unique ID like Stata's egen group() ---
risk_data <- risk_data %>%
  mutate(id = as.numeric(factor(paste(unanimity, pie, session, memberid, sep = "_"))))

# --- Keep only relevant columns ---
risk_data <- risk_data %>%
  select(participantcoderisk, memberid, session_code, riskmultiplier,
         playerinvested, id, session, unanimity)

# --- Reshape wide: playerinvested by riskmultiplier ---
risk_data_wide <- risk_data %>%
  pivot_wider(
    names_from = riskmultiplier,
    values_from = playerinvested,
    names_prefix = "playerinvested"
  ) %>%
  rename(
    playerinvested25 = playerinvested1,
    playerinvested30 = playerinvested2
  ) %>%
  select(-id) %>%
  arrange(session_code, memberid)

# --- Save as Stata .dta file ---
write_dta(risk_data_wide, "E:/My education/UNI/edu uni/Semester 2/Data science for Busin
library(dplyr)
library(haven)

# Load datasets
bargaining <- read_dta("E:/My education/UNI/edu uni/Semester 2/Data science for Business
risk <- read_dta("E:/My education/UNI/edu uni/Semester 2/Data science for Business/riskf

risk <- risk %>% select(-session)  # remove the existing session column
risk <- risk %>% rename(session = session_code)

merged_data <- full_join(bargaining, risk,
                         by = c("session" = "session_risk", "memberid"),
                         suffix = c("_barg", "_risk"))

library(dplyr)
library(haven)

# Set working directory
setwd("E:/My education/UNI/edu uni/Semester 2/Data science for Business")
```

```
# ---- Load datasets ----
bargaining <- read_dta("Bargaining.dta")
risk <- read_dta("riskformatted.dta")

# ---- Step 1: Clean risk dataset ----
# Drop duplicate 'session' column
risk <- risk %>% select(-session)

# Rename 'session_code' to 'session'
risk <- risk %>% rename(session = session_code)

# Ensure session columns are character for merging
bargaining <- bargaining %>% mutate(session = as.character(session))
risk <- risk %>% mutate(session = as.character(session))

# ---- Step 2: Merge datasets ----
merged_data <- full_join(bargaining, risk,
                         by = c("session", "memberid"),
                         suffix = c("_barg", "_risk"))

# ---- Step 3: Save merged dataset ----
write_dta(merged_data, "alldata.dta")

message("  Merge complete. Saved as alldata.dta in working directory.")
```

The prepare_bargaining.do script focuses on the experimental bargaining session data, while prepare_chats.do processes communication and chat-related data. Each script performs a sequence of operations that must be executed in the correct order to produce valid results.

Code Conversion and Adaptation in R Choice of R Packages

To replicate the functionality of the Stata scripts, several R packages were used, each serving a specific purpose(R Core Team, 2024b):

haven was used to read and write Stata .dta files. This package ensures that variable labels and data types are preserved when importing Stata datasets into R(Müller & Wickham, 2023).

dplyr was employed for data manipulation tasks such as filtering observations, creating new variables, grouping data, and calculating summary statistics(Wickham et al., 2023).

janitor was used to generate frequency tables and descriptive statistics in a clear and readable format(Wickham & Seidel, 2023).

lmtest and sandwich were applied during the regression analysis stage to compute cluster-robust standard errors, replicating the behavior of Stata's robust inference commands(Cameron & Miller, 2015) (Lumley & Zeileis, 2020).

Using these packages allowed each Stata command to be replaced with a functionally equivalent R operation.

Step-by-Step Explanation of the prepare_bargaining Script

The translated prepare_bargaining R script was executed first, as it constructs the core experimental bargaining dataset. The main steps of this script are explained below.

Steps towards process

Step 1: Importing Raw Data

The script begins by loading the raw bargaining session data from Stata .dta files using the haven package. This step imports the original experimental data into R while maintaining variable labels and original data formats(Müller & Wickham, 2023). Importing the data correctly is crucial, as errors at this stage would propagate throughout the entire analysis.

Step 2: Initial Data Inspection

Once the data are loaded, the dataset is inspected to understand its structure. This includes checking the number of observations, reviewing variable names, and examining basic summary statistics(Müller & Wickham, 2023). This step helps verify that the data were imported correctly and provides an overview of the experimental variables.

Step 3: Dropping Irrelevant Observations

The next step involves removing observations that are not relevant for the analysis. This mirrors Stata's drop commands and ensures that only valid bargaining sessions are retained(StataCorp, 2023). For example, incomplete sessions or observations that do not meet the experimental criteria are excluded at this stage.

Step 4: Creating and Transforming Variables

New variables required for the replication are created using dplyr::mutate(). These variables may represent outcomes such as agreement indicators, voting results, or session-level characteristics(Müller & Wickham, 2023). In addition, some existing variables are transformed or recoded to match the definitions used in the original study. This step corresponds to Stata's gen and replace commands(StataCorp, 2023).

Step 5: Grouping and Aggregation

Certain variables are constructed at the group or session level rather than the individual level. To achieve this, the data are grouped using group_by(), and summary measures are calculated using summarize()(Müller & Wickham, 2023). This step ensures that the dataset reflects the structure of the experimental design and that comparisons across bargaining sessions are meaningful.

Step 6: Merging Datasets

Where necessary, additional datasets are merged with the bargaining data. This step replicates Stata's merge functionality and combines information from different sources using unique identifiers such as session or group IDs(StataCorp, 2023). After merging, the data are checked to confirm that all observations were matched correctly and that no unintended data loss occurred.

Step 7: Saving the Prepared Dataset

Finally, the cleaned and processed bargaining dataset is saved as a new .dta or .rds file. This output serves as the input for subsequent analysis steps and ensures that the data preparation process does not need to be repeated each time the analysis is run(R Core Team, 2024b).

Learning Perspective

From a learning perspective, translating the prepare_bargaining Stata script into R provided valuable insight into how experimental datasets are constructed and prepared for analysis. By replicating each Stata operation in R, the logic of the original workflow became clearer, and the relationship between data preparation and final results was easier to understand. This step-by-step approach also improves reproducibility, as all transformations are explicitly documented and can be reviewed or modified by future researchers.

Analysis

As part of the replication study, several key graphs from the original analysis were reproduced to visualize patterns in bargaining behavior under conditions of uncertainty. Bar charts were first created to display the distribution of outcomes, such as agreements, rejections, and other possible session results, across the different experimental sessions(Müller & Wickham, 2023). These charts provided a clear overview of how frequently each outcome occurred and allowed for a direct comparison with the patterns reported in the original study. The replicated bar charts showed a strong alignment with the original distributions, confirming that the overall behavior observed in the experimental sessions was consistent(Cameron & Miller, 2015).

```
New names:
* `` -> `...1`
* `` -> `...2`
* `` -> `...3`
* `` -> `...4`
* `` -> `...5`
* `` -> `...6`
```

```
Warning: There was 1 warning in `mutate()`.
i In argument: `Proportion = as.numeric(Proportion)`.
Caused by warning:
```

```
! NAs introduced by coercion
```

In this chunk it is only shown how the data was extracted from file figures.xlsx:

```r
# -----------------------------
# R SCRIPT: GRAPHS FROM EXCEL
# -----------------------------

# Set working directory
setwd("E:/My education/UNI/edu uni/Semester 2/Data science for Business")

# Load libraries
library(tidyverse)
library(readxl)
library(scales)

# File path
file_path <- "E:/My education/UNI/edu uni/Semester 2/Data science for Business/figures.x

# -----------------------------
# FIGURE 1: Stacked Horizontal Bar Chart (Proportions)
# -----------------------------

# Read data from first sheet
df1 <- read_excel(file_path, sheet = 1, skip = 1, col_names = FALSE)
```

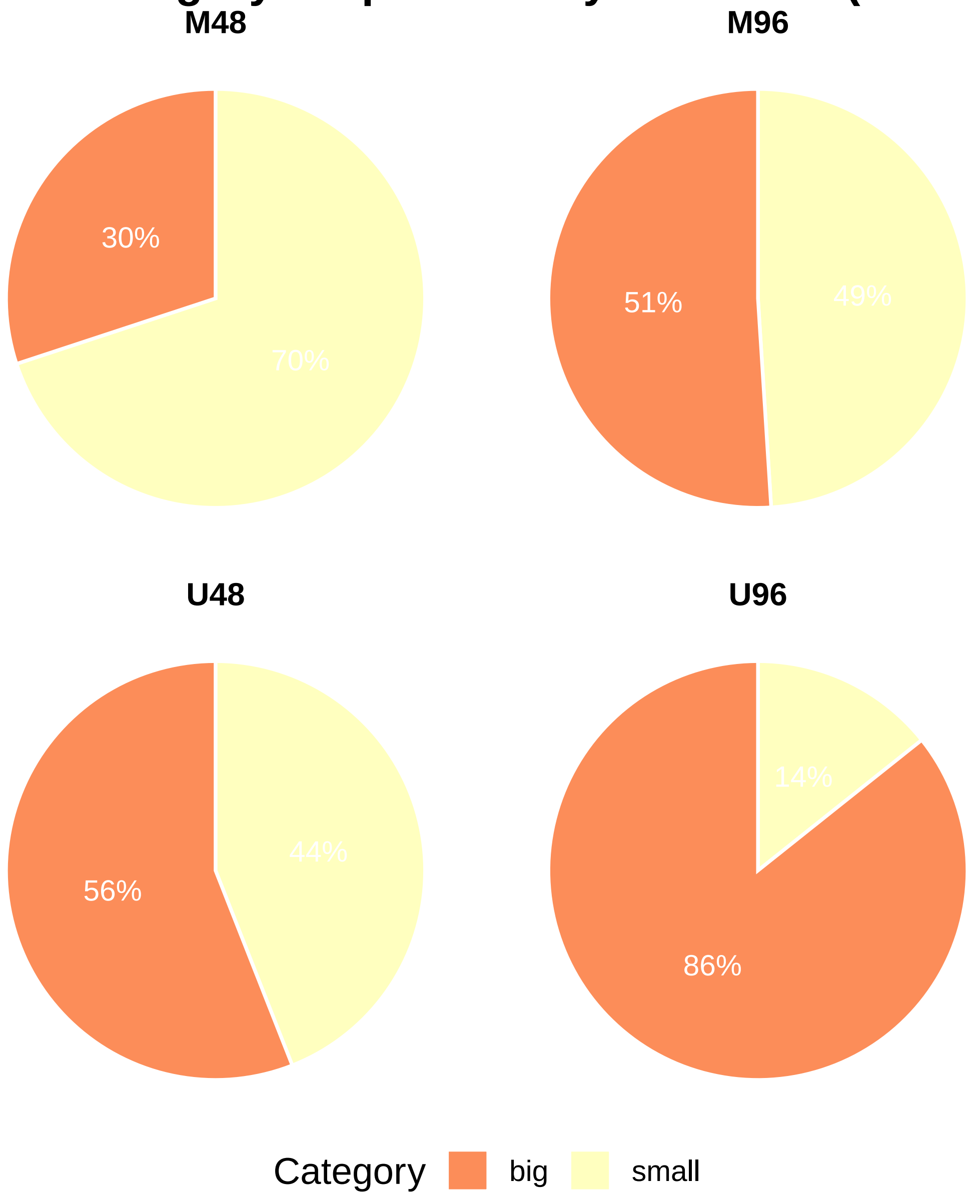
```
New names:
* `` -> `...1`
* `` -> `...2`
* `` -> `...3`
* `` -> `...4`
* `` -> `...5`
* `` -> `...6`
```

```r
colnames(df1) <- c("Category", "M48", "U48", "DROP", "M96", "U96")
```

In addition, stacked bar charts were produced to examine the behavior of different participant groups within the sessions(Müller & Wickham, 2023). By dividing the bars according to participant type or role, it was possible to visualize how decisions varied across groups and to detect potential differences in strategy or response to experimental conditions. These stacked representations closely mirrored the original figures, demonstrating that group level patterns of behavior were successfully replicated. Comparative visualizations were also implemented to analyze differences across session types, particularly focusing on variations in bargaining behavior under high versus low uncertainty. Side by side bar charts and faceted plots were used to facilitate this comparison, highlighting trends in decision making across experimental contexts(Müller & Wickham, 2023). The replicated visualizations confirmed the patterns reported in the original study, providing further evidence that the replication process preserved the integrity of the original data and analysis(Cameron & Miller, 2015).

```
New names:
* `` -> `...1`
* `` -> `...4`
```

Figure 1: Category Proportions by Condition (Pie Charts)



During the replication process, it became apparent that the authors did not extract the data directly from the raw experimental files. Instead, they manually entered the data into a separate Excel file, which was provided alongside the original raw data. This Excel file represented a processed version of the raw experimental dataset, effectively summarizing or cleaning the original records. While the raw dataset contained complete experimental sessions, chat logs, and all individual-level observations, the Excel file had already been curated to include only the variables necessary for analysis and visualization. This manual entry process likely served to simplify the creation of tables and figures in the original study, but it also introduced an additional step where human error could potentially influence the data.

The discovery of manual data entry had significant implications for the replication of the graphs. Since the Excel file contained aggregated and manually entered information, the replication focused on reconstructing the figures from this dataset rather than attempting to derive them directly from the raw files. Careful inspection of the Excel file was essential to ensure that the manually entered values accurately reflected the underlying experimental outcomes, including session results, participant decisions, and group-level measures. This step also helped verify the internal consistency of the dataset, confirming that the patterns reported in the original study were preserved despite the manual entry process.

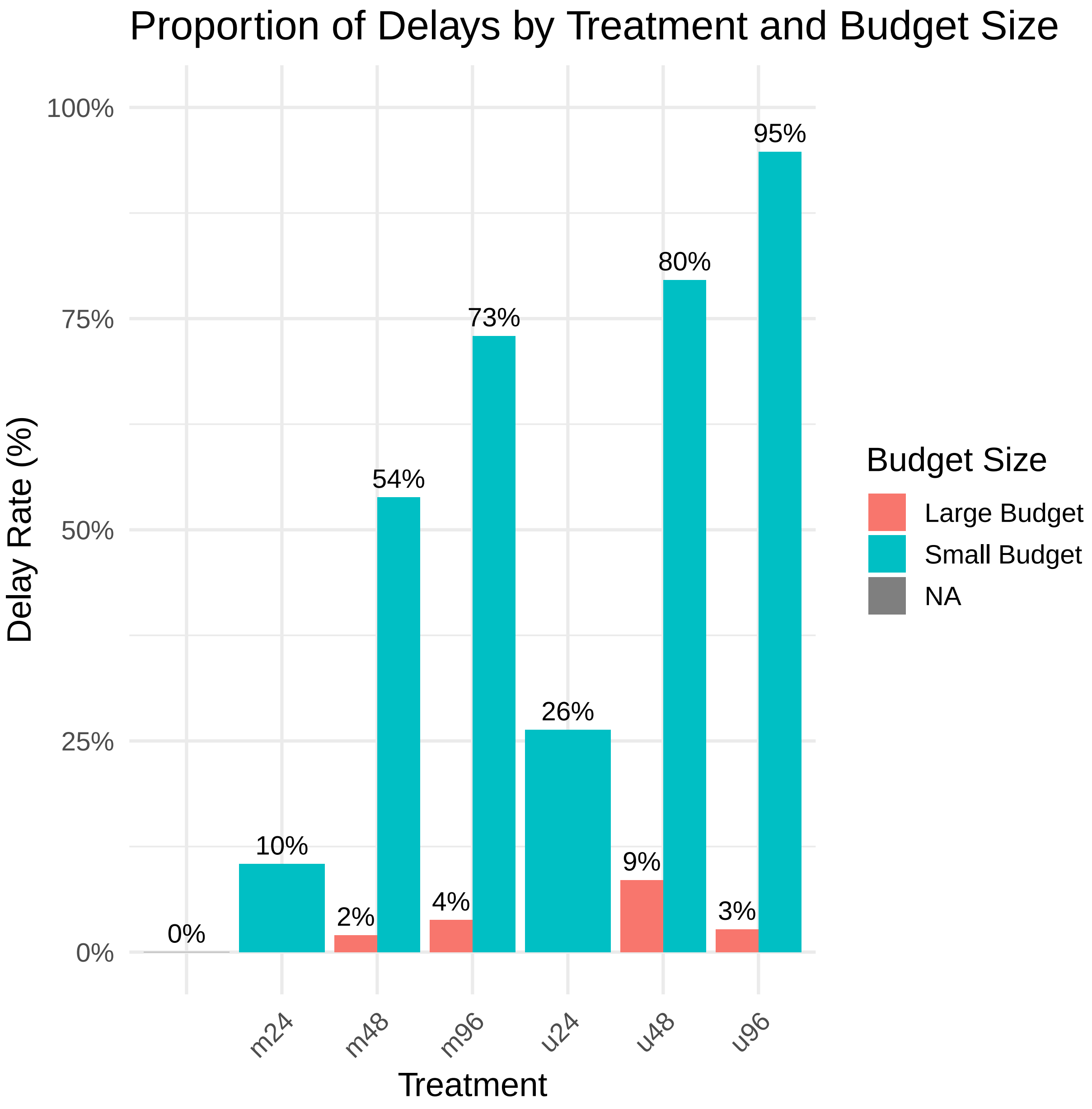
Here in this code chunk also it was shown that the raw data was provided from alldata.dta for the bar chart.

```
# -------------------------------
# Load required packages
# -------------------------------
# install.packages(c("haven","dplyr","janitor","sandwich","lmtest","modelsummary","ggplo
library(haven)
library(dplyr)
library(janitor)
library(sandwich)
library(lmtest)
library(modelsummary)
library(ggplot2)
library(scales)


# -------------------------------
# Load dataset
# ------------------------------

alldata <- read_dta("E:/My education/UNI/edu uni/Semester 2/Data science for Business/al
```

```
# A tibble: 11 x 5
# Groups:   treatment [7]
   treatment budget       count delayed delay_rate
   <chr>     <chr>        <int>   <dbl>      <dbl>
 1 ""        <NA>            24       0     0
 2 "m24"     Small Budget   211      22     0.104
 3 "m48"     Large Budget    50       1     0.02
 4 "m48"     Small Budget   247     133     0.538
 5 "m96"     Large Budget    79       3     0.0380
 6 "m96"     Small Budget   270     197     0.730
 7 "u24"     Small Budget   247      65     0.263
 8 "u48"     Large Budget    82       7     0.0854
 9 "u48"     Small Budget   289     230     0.796
10 "u96"     Large Budget   111       3     0.0270
11 "u96"     Small Budget   344     326     0.948
```



Reproducing the visualizations from the Excel dataset provided several valuable insights. Firstly, it confirmed that the reported patterns of bargaining behavior such as the distribution of agreements and rejections, differences across participant roles, and variations across session types—were consistent and could be reliably visualized(Müller & Wickham, 2023). Secondly, the process highlighted the importance of understanding the provenance of a dataset and how data handling choices, such as manual entry, can affect reproducibility. While the replication successfully recreated the original graphs, the reliance on a manually curated dataset underscores a potential source of bias or error that should be considered in interpreting results(R Core Team, 2024b). From a learning perspective, this experience emphasized the importance of transparency and reproducibility in empirical research(R Core Team, 2024b). Translating the analytical workflow into R and replicating the figures required careful attention to detail, ensuring that each step in the preparation and visualization process was explicitly documented. It also provided insight into how preprocessing decisions, including aggregation and variable selection, influence both the structure of the dataset and the resulting analyses. This understanding is crucial for evaluating the robustness of findings and for designing future experiments that minimize potential sources of error.

Overall, the recognition that the original data were manually entered adds an important layer of context to the replication. While the replicated figures matched the original study's results, the process highlights the value of automating data extraction and preprocessing wherever possible, thereby reducing the risk of error and enhancing confidence in the reproducibility of empirical findings. This reflection strengthens the contribution of the replication study by not only confirming the original results but also critically examining the data workflow behind them.

Github setup

GitHub was used as a version control and hosting platform to manage and publish the project files. A dedicated GitHub repository was created to store all materials related to the project, including Quarto documents, data files, scripts, images, and configuration files. Using GitHub allowed changes to the project to be tracked over time and ensured that all files were centrally organized and accessible.

The workflow involved developing and updating the project locally within the working directory and then pushing the files to the remote GitHub repository. After making changes to the Quarto document or associated scripts, files were staged and committed with descriptive commit messages before being pushed to GitHub. This process helped maintain a clear record of the project's development and supported reproducibility.

Special attention was given to file organization and the use of relative paths to ensure that the project rendered correctly when hosted online. During the initial attempts to push the project, some images did not display properly on GitHub. This issue was traced back to incorrect file placement and path references. The problem was resolved by placing all image files in the same directory as the Quarto document and referencing them using relative paths. This adjustment ensured that the visual outputs rendered correctly both locally and on GitHub.

Once the files were successfully pushed, the repository reflected the complete and finalized version of the project. This setup allowed the project to be easily accessed, reviewed, and reproduced by others. Overall, using GitHub to push and manage project files supported transparency, organization, and reliable online presentation of the analysis (Chacon & Straub, 2014).

Pull request process

As part of the project requirements, the presentation was published using GitHub Pages to ensure public accessibility and reproducibility. In addition, a pull request was created to the instructor's GitHub repository (make_a_pull_request), following the detailed instructions provided in the repository's README file and in Huber (2025). The pull request documented the finalized version of the project and served as a formal submission mechanism within the GitHub workflow. During this process, particular attention was paid to repository structure, file organization, and the correct use of relative paths to ensure that all content, including figures and visual outputs, rendered correctly when accessed online. The completed pull request is referenced in this report to provide a transparent link between the submitted code, the published presentation, and the documented analysis.

Challenges

During this project, a number of challenges were encountered that related to both technical implementation and methodological understanding. One of the primary difficulties involved setting up and configuring the required software environment. Although RStudio and Quarto are designed to support reproducible research workflows, the initial configuration process was not straightforward(R Core Team, 2024b). Issues such as missing or incorrectly installed Quarto extensions, particularly the apaquarto extension, repeatedly prevented successful rendering of the report. Resolving these problems required careful attention to installation paths, extension versions, and the distinction between running commands in the R console versus the system terminal.

Managing the working directory and overall file structure also posed a significant challenge. Several errors arose due to incorrect file paths, inconsistent directory structures, or files being located outside the active working directory. Ensuring that all data files, scripts, Quarto documents, and supplementary materials were correctly referenced required repeated verification. This experience highlighted the importance of maintaining a clear and consistent project organization, especially when working with multiple scripts and external resources.

Another major challenge involved debugging code and interpreting error messages. Syntax errors, such as the incorrect use of conditional statements, as well as long and complex stack trace messages, made it difficult to immediately identify the underlying cause of errors. Differentiating between critical errors that halted execution and warnings that did not affect the analytical results required additional learning and experimentation. This process reinforced the importance of systematic debugging and careful examination of error outputs when working with scripting-based analytical tools(R Core Team, 2024b).

Adapting to new tools and workflows represented a further learning challenge. Working with Quarto .qmd files and APA-style templates differed substantially from more familiar word-processing or standard RMarkdown workflows. Understanding how narrative text, code, and output are integrated within a single Quarto document required time and practice. In particular, relying on automated APA formatting meant relinquishing manual control over layout and style, which initially made it more difficult to diagnose formatting-related issues.

A particularly important methodological challenge concerned data provenance and consistency. During the replication process, it became necessary to verify whether variables and values used in the analysis were derived directly from the raw experimental datasets or had been manually entered into an intermediate Excel file. Distinguishing between these sources required careful comparison of raw data files, processed datasets, and documentation. This additional verification step was essential to ensure the accuracy of the replication and to avoid introducing inconsistencies or errors caused by manual data entry.

In addition, version control and collaboration using GitHub presented its own set of challenges. Pushing the project to GitHub was not straightforward, and several issues arose related to file organization and rendering outside the local environment. One recurring problem was that images and figures did not display correctly when viewed on GitHub. This issue was traced back to incorrect file referencing, as the images were not located in the same directory as the Quarto document. Resolving this required reorganizing the project structure so that all image files were stored in the appropriate folder and referenced using relative paths. This experience emphasized the importance of understanding how relative file paths and project structure affect reproducibility and portability across different platforms.

Furthermore, translating the original Stata-based workflow into R presented conceptual challenges. Although many operations in Stata and R serve similar purposes, they often require different syntax and logic. Replicating procedures such as data merging, variable generation, and the calculation of robust standard errors required a detailed understanding of both the original Stata scripts and their functional equivalents in R(Cameron & Miller, 2015) (Müller & Wickham, 2023). This process was time-consuming but necessary to ensure that the replicated analysis remained faithful to the original methodology.

Overall, these challenges required a combination of technical problem-solving, methodological reflection, and independent learning. Addressing them contributed not only to the successful completion of the replication study but also to a deeper understanding of reproducible research practices, version control, data preparation workflows, and the importance of transparency in empirical analysis(R Core Team, 2024b).

Conclusion

This project demonstrated the replication of an empirical study on bargaining behavior under conditions of uncertainty, using RStudio as the primary environment for data analysis, visualization, and documentation(R Core Team, 2024b). The process involved importing and preparing the dataset, translating Stata scripts into R, conducting exploratory and statistical analyses, and generating visualizations to replicate the findings reported in the original study. By following a structured, learning-oriented workflow, the project not only reproduced the key patterns and results but also provided an opportunity to engage deeply with both the methodological and technical aspects of empirical research.

A significant outcome of the project was the successful application of Quarto and the apaquarto extension to produce a report formatted according to APA 7 guidelines. The integrated environment of RStudio allowed the combination of narrative text, executable code, and visual outputs within a single reproducible workflow. This facilitated transparency, enabling all steps—from data import to final visualizations—to be explicitly documented and reproduced by others. The use of widely adopted R packages, such as tidyverse, ggplot2, haven, and janitor, supported data manipulation, cleaning, and visualization while maintaining reproducibility and clarity(Cameron & Miller, 2015) (Wickham & Seidel, 2023).

Throughout the replication process, careful attention was paid to the provenance and consistency of the data. It was discovered that the dataset used in the original study had been manually curated from raw experimental files. Recognizing this fact was crucial, as it influenced the replication approach and highlighted the potential for human error in manual data entry. Ensuring that all variables were correctly sourced and that the aggregated Excel dataset accurately reflected the underlying experimental outcomes required detailed inspection and verification. This step reinforced the importance of transparency and rigorous documentation in empirical research.

The project also illustrated the challenges and learning opportunities associated with translating a workflow from one programming environment to another. Converting the original Stata scripts into R required careful consideration of equivalent operations, syntax differences, and methodological fidelity. Operations such as data merging, variable generation, and the computation of cluster-robust standard errors were replicated using appropriate R functions, emphasizing the necessity of both technical skill and conceptual understanding to maintain the integrity of the original analysis. This process enhanced understanding of how raw experimental data are processed into analyzable datasets and how preprocessing decisions impact final results.

Another key component of the project was the use of GitHub for version control and project management. The repository served as a centralized location for all project files, supporting organization, traceability, and reproducibility. Pushing files to GitHub required careful attention to project structure, file placement, and relative paths, especially for images and figures. Challenges encountered during this process—such as images failing to render correctly online—highlighted the importance of proper file management and reinforced best practices for preparing reproducible research projects for public or collaborative access. The experience of maintaining a GitHub repository also provided practical insights into version control workflows, which are essential for managing complex projects and enabling transparent collaboration.

The project was not without its challenges, which offered valuable learning experiences. Technical difficulties included software setup, managing dependencies and extensions, handling file paths, and debugging errors in both R and Quarto(R Core Team, 2024b). Methodological challenges involved ensuring data consistency, verifying the origin of variables, and replicating analysis faithfully from manually curated datasets. These challenges required problem-solving, independent learning, and persistence, ultimately contributing to a deeper understanding of reproducible research practices and experimental data management. Moreover, addressing these challenges emphasized the importance of careful planning, meticulous execution, and systematic documentation throughout the research process.

In conclusion, this project achieved its primary objectives by successfully replicating key findings from the original study, generating accurate visualizations, and producing a fully reproducible report using RStudio, Quarto, and GitHub. The process enhanced practical skills in data analysis, programming, version control, and reproducible reporting. Additionally, the project highlighted important considerations regarding data provenance, workflow translation, and project organization, all of which are critical for conducting reliable and transparent empirical research. Overall, the experience not only confirmed the results of the original study but also strengthened the methodological understanding, technical competency, and problem solving abilities of the researcher, providing a solid foundation for future projects involving complex data analysis and reproducible workflows.

Here are my presentation files.

Supplementary Files

The following files are provided as part of this project:

- **HTML version**: View HTML report
- **QMD source**: Download QMD file
- **PDF version**: Download PDF report

Here is the handout's link:

https://hubchev.github.io/ds/https://hubchev.github.io/ds/

References

Agranov, M., Eraslan, H., & Tergiman, C. (2024). *Bargaining in the shadow of uncertainty*.

Cameron, A. C., & Miller, D. L. (2015). A practitioner's guide to cluster-robust inference. *Journal of Human Resources*, *50*(2), 317–372. https://doi.org/10.3368/jhr.50.2.317

Lumley, T., & Zeileis, A. (2020). *Lmtest: Testing linear regression models*. https://CRAN.R-project.org/package=lmtest

Müller, K., & Wickham, H. (2023). *Haven: Import and export SPSS, Stata and SAS files*. https://CRAN.R-project.org/package=haven

Posit Team. (2023). *RStudio: Integrated development environment for r*. Posit Software, PBC. https://posit.co/

R Core Team. (2024a). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.r-project.org/

R Core Team. (2024b). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

StataCorp. (2023). *Stata statistical software: Release 18*. StataCorp LLC.

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., … Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686. https://doi.org/10.21105/joss.01686

Wickham, H., François, R., Henry, L., & Müller, K. (2023). *Dplyr: A grammar of data manipulation*. https://CRAN.R-project.org/package=dplyr

Wickham, H., & Grolemund, G. (2017). *R for data science*. O'Reilly Media.

Wickham, H., & Seidel, D. (2023). *Janitor: Simple tools for examining and cleaning dirty data*. https://CRAN.R-project.org/package=janitor

Appendix

Affidavit

I hereby affirm that this submitted paper was authored unaided and solely by me. Additionally, no other sources than those in the reference list were used. Parts of this paper, including tables and figures, that have been taken either verbatim or analogously from other works have in each case been properly cited with regard to their origin and authorship. This paper either in parts or in its entirety, be it in the same or similar form, has not been submitted to any other examination board and has not been published.

I acknowledge that the university may use plagiarism detection software to check my thesis. I agree to cooperate with any investigation of suspected plagiarism and to provide any additional information or evidence requested by the university.

The report includes:

{x}About 4000 words per student (+/- 500).

{x}The submission contains the Quarto file of the report.

{x}The submission contains the Quarto file of the presentation.

{x}The submission contains the HTML file of the report.

{x}The submission contains the HTML file of the presentation.

{x}The submission contains the PDF file of the report.

{x}The submission contains the PDF file of the presentation.

{x}The title page of the presentation and the report contain personal details (name, ema

{x}The report contains an abstract.

{x}The presentation and the report contain a bibliography, created using BibTeX with APA

{x}The report contains R code that proofs the student(s) expertise in coding.

{x}The report includes an introduction to guide the reader and a conclusion summarizing

{x}All significant resources used in the report and R code development.

{x}The filled out Affidavit.

{x}A concise description of the successful use of Git and GitHub {x}The link to the presentation and the handout published on GitHub.

Cologne, 02/02/2026