Data Science for Business – Replication Project

Christian Faruk Rodríguez Casas
rodriguez_casas.christian@stud.hs-fresenius.de
ID: 400791457
Cologne, Germany

Table of contents

Correspondence concerning this article should be addressed to Christian Faruk Rodríguez Casas, Email: rodriguez_casas.christian@stud.hs-fresenius.de

Data Science for Business – Replication Project

Abstract

The process required us to pay close attention to how the authors structured their simulation, modeled their data, and understood the market dynamics. While we successfully replicated most outputs, a few figures didn't match exactly due to limitations in the replication data and differences in how certain functions behave in R compared to Stata.

Beyond the technical challenges, the replication project helped us not only deepen our coding knowledge, but also to understand how algorithmic pricing influences competitive outcomes, especially the tension between learning and collusion. Furthermore, it gave us a clearer view of how to structure academic research with the aid of R, insight that will be valuable when writing our master's thesis.

The following sections walk through our workflow, key coding decisions, and the main insights we gathered throughout the process.

1 Introduction

At first glance, replicating an academic paper might seem easy on the surface: choose a topic, pick a paper, review its graphs and tables, assess its difficulty, analyze its replication package, data sources, code, match the results, and publish. But in reality, it rarely works out that cleanly.

This report offers a clear, transparent, and honest account of our attempt to replicate the paper *Competition in Pricing Algorithms* by Brown and MacKay (2023), a study that explores how businesses that use AI-based pricing tools can unintentionally (or intentionally) reduce competition over time. In the authors' words the study aims to:

> "illustrate how pricing algorithms can generate supracompetitive prices through novel, non-collusive mechanisms. Frequency, commitment, and asymmetry in pricing technology allow firms to support higher prices in competitive (Markov perfect) equilibrium" (Brown & MacKay, 2023, p. 2)

2 Motivation

Our main motivation going into this replication project was to better understand how could we with a software replicate a paper, and most importantly, assuming a challenge that would mean do some reverse-engineering to understand a code that up until that point we had never seen before, and more importantly, how to recreate such an analysis ourselves using a tool we were comfortable with: R. We weren't interested in just clicking through a replication package and graphing out the first graph we found. We wanted to internalize what the original authors did, why they did it, and how their findings were shaped by the structure of their data and the tools they used.

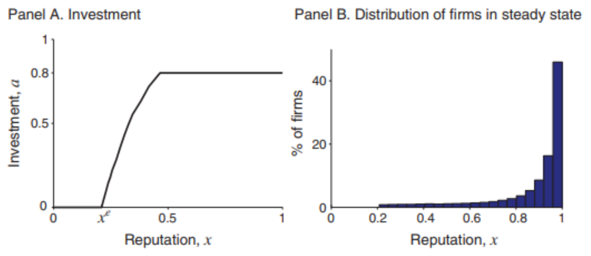
3 Selecting a Paper to Replicate

One of our first (and perhaps most important) decisions was choosing which paper to replicate. Without an advanced economics background, we aimed for a project that was manageable but still meaningful. After careful consideration, we narrowed it down to two potential options:

1. ***A Reputational Theory of Firm Dynamics*** (Board & Meyer-ter-Vehn, 2022)
2. ***Competition in Pricing Algorithms*** (Brown & MacKay, 2023)

Figure 1

Consumers Observe Firms' Investment

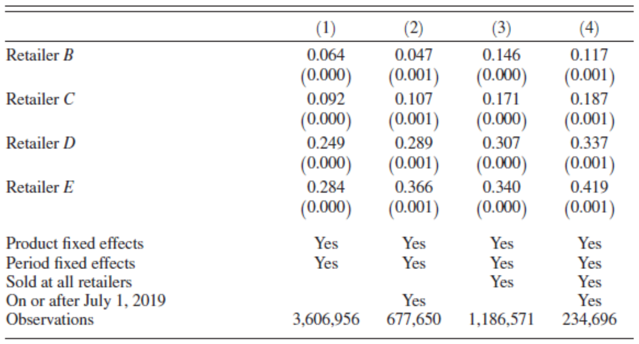


Out of these two papers we had to make a choice, once again based purely on intuition. On one hand, the first paper seemed more accessible, with not that complicated graphs (see Figure 1 from Board and Meyer-ter-Vehn (2022), not that many data sources and with (to our knowledge so far), a replication package that could easily be replicated. In contrast, the second paper looked more challenging, complex, with multiple data sources, graphs and tables, see Figure 2 made from Brown and MacKay (2023) with a new software and programming language that we did not understand – Stata.

We initially chose the article written by Board and Meyer-ter-Vehn (2022) but soon understood that it was attractive and straightforward only on the surface because it used simulated data, one aspect that over complicated the replication process. However, after reviewing the replication package and the ***README*** file (Open) for the second paper "Competition in Pricing Algorithms" , which studies the competitive effects of pricing algorithms using proprietary data from online drugstores, we noticed that, despite its complexity, size, multiple sources and the use of Stata scripts, Brown and MacKay

Figure 2

Price Differences for Identical Products Relative to Retailer A



	(1)	(2)	(3)	(4)
Retailer *B*	0.064	0.047	0.146	0.117
	(0.000)	(0.001)	(0.000)	(0.001)
Retailer *C*	0.092	0.107	0.171	0.187
	(0.000)	(0.001)	(0.000)	(0.001)
Retailer *D*	0.249	0.289	0.307	0.337
	(0.000)	(0.001)	(0.000)	(0.001)
Retailer *E*	0.284	0.366	0.340	0.419
	(0.000)	(0.001)	(0.000)	(0.001)
Product fixed effects	Yes	Yes	Yes	Yes
Period fixed effects	Yes	Yes	Yes	Yes
Sold at all retailers			Yes	Yes
On or after July 1, 2019		Yes		Yes
Observations	3,606,956	677,650	1,186,571	234,696

(2023) offered an explanation detail that was (for us), an incredible asset because it clearly explained what each folder, function, variable, and data source meant.

The package also includes multiple well-documented Stata scripts, such as ***07_summary_stats.do*** (Open), which are used to generate the tables and figures in the paper. Initially, we found the concepts in this paper quite complex and the graphs and tables (see Figure 2 and Figure 3) more challenging to replicate but after comparing the replication packages from both papers we decided to change course and ultimately, we chose to proceed with our "Plan B": **Competition in Pricing Algorithms** (Brown & MacKay, 2023). Despite its complexity, it uses real, proprietary data and provides thoroughly organized and transparent replication package, factors that aligned better with our objectives and offered a more enriching learning experience.

4 The Choice

The deciding factor wasn't complexity, but substance. Brown and MacKay (2023)'s paper doesn't rely on simulations; it analyzes real price data and market behavior in a way that is directly relevant to both policy and business. The replication package was also extremely well-documented, with clear connections between scripts and final tables or figures. That transparency allowed us to dig deep into how the authors structured their workflow. Thus, offering us the opportunity to not just "replicate" their analysis, but to understand and rebuild it using an entirely different statistical language: R.

Despite some early challenges, like matching plot styles, interpreting Stata commands, and keeping R from crashing, we successfully replicated key outputs from the paper, including dynamic price charts and detailed summary statistics. In the process, we not only learned to navigate the technical side of replication but also came to appreciate the value of reproducible research and cross-platform translation in empirical economics.

5 Procedure

After we selected Brown & MacKay's Competition in Pricing Algorithms paper, our first step was to break down the project and its graphs into six phases:

1. understanding the paper and their concepts.
2. Dividing the work between our initial team of 3 people.
3. Reviewing the replication package and its contents with the help of the ***README*** file(Open).
4. Understand the ***07_summary_stats.do*** (Open) file and its code.
5. Translating Stata code into R and associate that to a graph or table in particular.
6. Recreating the visualizations and summary statistics.

We knew from the start this would require both technical skills and a lot of trial and error, especially since neither of us had worked with Stata before, had a basic understanding of R and we had never translated code across programming languages.

We began by reverse-engineering the logic of each .do file in the Stata script package. This means:

1. Reading line by line.
2. Running small chunks of code.
3. Checking outputs.
4. Figuring out how each step connected to a table or graphs in the paper.

We relied heavily on documentation, forums, AI and our own instincts to understand what each function was doing, and how we might translate that behavior into tidyverse-based R code. This was time-consuming but necessary to achieve the desired outcome – being able to successfully replicate a paper with RStudio and push those results to Github Open our Landing Page.

Once we had a handle on the logic, we gradually moved into rebuilding the pipeline in R, organizing it into scripts that mirrored the structure of the original Stata files.

Publishing our work to GitHub from the start also gave the project a sense of transparency and progress. It allowed us to share issues, track changes, and build the habit of clean, replicable workflows, something we'll definitely carry forward into future research work.

6 Challenges and Issues when coding

Our first and most important challenge was not coding based; it was teamwork and that meant transforming teamwork into real progress. Our group included Tony Osei, Vansh Patni and me – Christian F. Rodríguez Casas. The team dynamic between Tony and me was great from the beginning. Since we were both physically in Cologne, it was easier to meet and coordinate our tasks. But it wasn't the same with Vansh – he was not physically in Germany and due to time differences, it was complicated to coordinate responsibilities.

This led us to work more independently and create a shared landing page. It allowed us to push our work to Github and connect each other's work through links on a single platform. As a result, we reduced dependency on Vansh and his inputs. This idea proved to be one of our best, because it gave Tony and me a work structure where we could support each other, and at the same time we could work freely in our code and presentation. while still functioning as a team – smaller one, but maybe a better one.

The transition from Stata to R was not just a technical challenge, it often became a conceptual one. Many commands in Stata that take a single line would require more elaborate, layered steps in R. For example, some of the data manipulation in Stata involved implicit behaviors (like preserving labels, handling missing values silently, or reshaping data in-place) that don't work the same way in R. This meant we had to be extremely careful when translating steps: we couldn't just find a "Stata-to-R" equivalent, we had to fully understand what each line was doing to the data and rebuild the logic from scratch.

Another recurring issue was replicating the visuals, especially the exact style and layout of graphs. For example, some of the figures in the original paper used very specific formatting (color schemes, axis scales, line styles) that were easy to produce in Stata but required custom tweaking in ggplot2 (Wickham, 2016). Getting the labels, axis limits, or breakpoints to match exactly took far more effort than expected. At one point, we spent hours trying to replicate a simple-looking price chart only to realize that the original Stata script was using a smooth conditional mean, while our R version was plotting raw data points.

In addition, our R environment sometimes struggled with memory issues when handling the full data, particularly when working with large .dta files. There were several instances where RStudio would freeze or crash, especially when we ran multiple nested mutate() and group_by() operations without optimizing memory usage.

Interpreting certain statistical procedures was another obstacle. For example, some regressions in the paper relied on built-in Stata routines that have no direct equivalent in R. In these cases, we had to dig into the documentation or find packages (like fixest, lmtest, or sandwich) that approximated the behavior as closely as possible. In a few cases, we had to make judgment calls on whether the small differences in coefficients were due to numerical precision or deeper structural issues in our code.

If we were asked to summarize the issues and challenges, we encountered throughout our journey it would be as follows:

1. Teamwork with somebody not available.

2. Interpreting the Stata scripts and mapping each section to its corresponding chart or table in the paper.

3. Managing GitHub publishing, dealing with data size limitations and the dilemma between using a public or private repository.

4. Troubleshooting project structure issues, such as realizing the index file was mistakenly placed outside the main working directory.

5. Handling R crashes,there were moments when R would freeze or become unresponsive; using Ctrl+C proved useful to "unstuck" it.

6. Fine-tuning visualizations,adjusting scaling, colors, and layout to match the presentation style in Brown and MacKay (2023), while ensuring the underlying data aligned correctly.

Despite these challenges, each problem we encountered ended up teaching us something new, whether about R's limits, statistical modeling, or just the importance of debugging with intention. The process didn't always feel linear or efficient, but it was genuinely instructive.

7 Replication Process

This section walks us through the steps taken in order to reproduce the **Figure 3**, ***Time series of prices across retailers***, found on page 118 of Brown and MacKay (2023) and the **Figure 4** ***Daily statistics for hourly price data***, found on page 117 of Brown and MacKay (2023).

But before elaborating and explaining in detail how our replication process unwrapped, I would like to introduce the tools and libraries used in our project:

7.1 Tools:

- **RStudio** (data analysis & visualization)

- **Quarto** (reproducible publishing)
- **GitHub Pages** (to share everything online)

7.2 Libraries:

-haven: this library created by Wickham, Miller, et al. (2025) allows R to read and write data files from statistical software like Stata, SPSS, and SAS.
-dplyr: Wickham, François, et al. (2025) provides a grammar of data manipulation, making it easy to filter, select, summarize, and transform data frames.
-ggplot: As stated by Wickham (2016) powerful and flexible system for creating static data visualizations using the grammar of graphics.
-janitor: Offers simple functions for cleaning and formatting data, especially column names and tabular summaries.

7.3 Understanding the Data and Folder Structure

I started by getting a feel for the structure of the replication package and the information shared by the authors in the README File. It came with a clear folder layout , data/, output/, programs/, and the main data source ***analysis_data.dta*** was just the entry point. We needed to analyze the main Stata script ***07_summary_stats.do*** (Open).

It became clear early on that the main .do file had all the info that we needed to replicate the main graphs and tables; we just had to understand it and connect it to sections inside the paper. I needed to understand and replicate what each helper script was doing, especially since many of them created intermediate data or reshaped variables in ways that were never fully documented. So, I built my R project and started working together with ChatGPT to try to understand and decode the Stata script.

7.4 Replicating Graphs

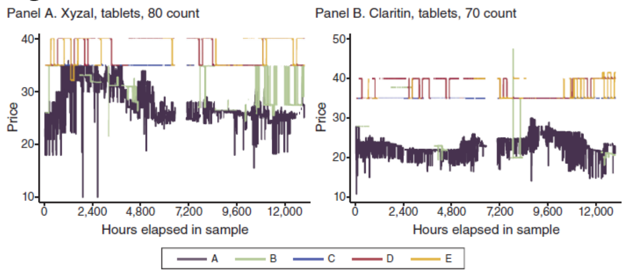
Replicating **Figure 3** took more patience than I expected, mostly because Stata's default plotting behavior handles multi-line graphs different to ggplot2. I reproduced both panels (one for Xyzal and the second one for Claritin) by coding with R and matching colors, axis limits, and layout details. To get the layout right, I used patchwork function **::plot_layout()**, which let me stack both plots in a way that replicates the original intended graph.

One small but important step was guaranteeing that the data was filtered out the same way the original authors did. That means filtering correctly by the product style required (ie. 80-count tablets in single packs) and removing any imputed prices. I also had to find a way to set the x-axis range using the maximum price to get clean, consistent marks.

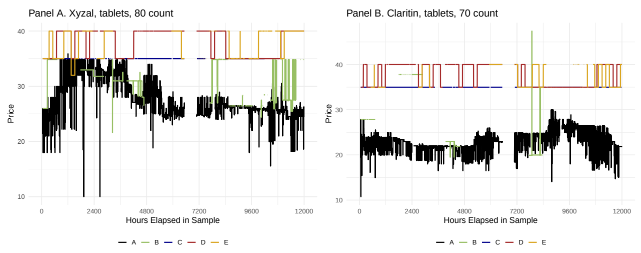
The graphs below provide a detailed comparison of the replication results through a side-by-side presentation of the original figure and our replication, allowing for a closer inspection of how closely our results align with the original analysis:

Figure 3, original graph ***"Time series of prices across retailers"*** (Brown & MacKay, 2023, p. 118):

Figure 3



Graph Replication:



Even though the final plots look straightforward, getting there took a lot of effort and trial and error, aligning font sizes, adjusting legends, and revisiting theme elements until they matched the original figures. I wrapped up the process with **ggsave()** inside a loop so I could reproduce the graphs later with consistent naming.

7.5 Replicating Tables

For **Figure 4**, which gives an overview of daily pricing activity by website, I built things up in two steps. First, I summarized the data at the product-date level to track price changes, daily averages, and whether a price change occurred. Then, I further aggregated key summary statistics like the number of products, share of products with price changes, and price percentiles to a website-date level.

To create a final summary per website, I grouped by them by website and took time-averaged values across all days. I also added a "Total" row by averaging metrics across all websites to reflect overall platform-level behavior. The resulting table closely replicates not only the layout in the original paper, but also its data and content.

I formatted the output using the function gt, transposing rows and columns so I could guarantee that the statistics appeared as row names and websites as columns, just like in the original table as seen in the comparison below:

Figure 4, original Table ***"Daily statistics for hourly price data"*** (Brown & MacKay, 2023, p. 117):

Figure 4

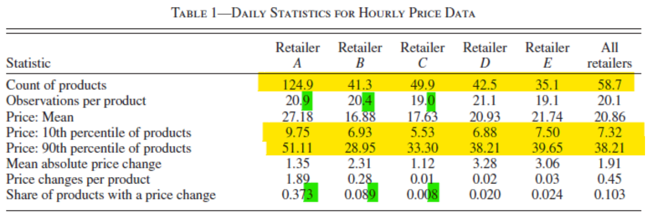


TABLE 1—DAILY STATISTICS FOR HOURLY PRICE DATA

Statistic	Retailer *A*	Retailer *B*	Retailer *C*	Retailer *D*	Retailer *E*	All retailers
Count of products	124.9	41.3	49.9	42.5	35.1	58.7
Observations per product	20.9	20.4	19.0	21.1	19.1	20.1
Price: Mean	27.18	16.88	17.63	20.93	21.74	20.86
Price: 10th percentile of products	9.75	6.93	5.53	6.88	7.50	7.32
Price: 90th percentile of products	51.11	28.95	33.30	38.21	39.65	38.21
Mean absolute price change	1.35	2.31	1.12	3.28	3.06	1.91
Price changes per product	1.89	0.28	0.01	0.02	0.03	0.45
Share of products with a price change	0.373	0.089	0.008	0.020	0.024	0.103

Table Replication:

Figure 5

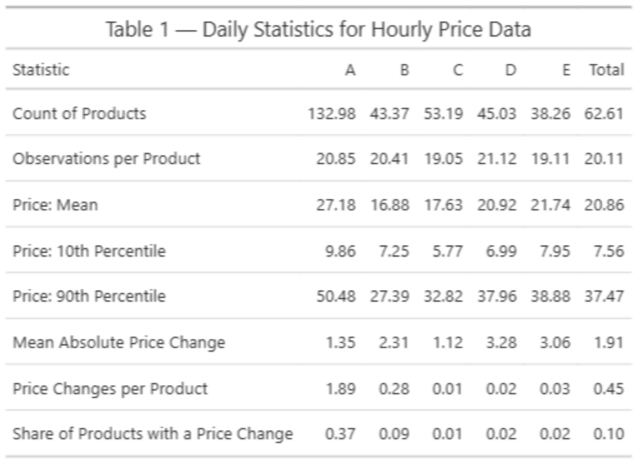


Table 1 — Daily Statistics for Hourly Price Data

Statistic	A	B	C	D	E	Total
Count of Products	132.98	43.37	53.19	45.03	38.26	62.61
Observations per Product	20.85	20.41	19.05	21.12	19.11	20.11
Price: Mean	27.18	16.88	17.63	20.92	21.74	20.86
Price: 10th Percentile	9.86	7.25	5.77	6.99	7.95	7.56
Price: 90th Percentile	50.48	27.39	32.82	37.96	38.88	37.47
Mean Absolute Price Change	1.35	2.31	1.12	3.28	3.06	1.91
Price Changes per Product	1.89	0.28	0.01	0.02	0.03	0.45
Share of Products with a Price Change	0.37	0.09	0.01	0.02	0.02	0.10

This structure improved readability and facilitated quick comparison between original and replication. While I didn't implement bootstrapped standard errors as in some of the paper's later tables, the descriptive statistics were closely replicated and aligned.

7.6 Debugging and Matching Outputs

One challenge that came up was getting my numbers to line up exactly with the ones in the paper. Even for basic averages or percentiles, Stata and R don't always behave the same , especially when it comes to missing values or rounding. I made sure to follow the same filtering steps as the original authors, including removing imputed prices, and used ***janitor::clean_names()*** early on to avoid any variable name mismatches.

Although I didn't replicate the regression models or bootstrapped standard errors from later sections of the paper, I was careful to match the structure, logic, and aggregation levels used in the descriptive tables. That way, the summary stats I report are calculated on the same footing and time frame as the original results.

Lastly, I invite the reader to examine the replication code in detail, for both Figure 3 (see code) and Figure 4 (see code), available in segmented chunks in the Appendix.

7.7 Successful Use of Git & GitHub

Throughout the project, we applied the standard fork–clone–edit–pull request workflow as outlined in the GitHub tutorial. We forked and cloned the Project Repository locally, created and worked from the main branch, and made iterative changes to key .qmd files — including (index.qmd), try.qmd(try.qmd), and Report.qmd (Report.qmd). Each change was committed with descriptive messages and pushed to GitHub. Although we did not submit a pull request to an upstream repository, the structure allowed us to simulate the full workflow and maintain clean version control. This approach reinforced reproducibility and collaborative standards in line with open-source research practices.

7.7.1 Making a pull request with Git and Github

To share my replication project with the professor, I followed the professors instructions for contributing to a public repository:

1. Forked the original repository make_a_pull_request provided by the professor to our own GitHub accounts.
2. Cloned the forked repository to our local computer using git clone.
3. Created a new branch named **add-Christian-Casas** using git switch -c.
4. Made changes to the ***I_am_a_data_scientist-md*** file, adding my name, my GitHub account and the paper I reproduced.
5. Staged and committed the changes with a clear commit message **Add Christian Faruk ROdriguez Casas to the list** using git add . and git commit -m.
6. Pushed the branch to GitHub using: **git push -u origin add-Christian-Casas**.
7. Submitted my changes for review on Github and Created succesfully the pull request.

8 Remaining Issues and Open Challenges

Even though I was able to replicate the main graph and achieve a close replication result to the table, a few things still could have been improved. Some parts of the Stata code are either undocumented or can only be speculated, like filters and other kinds of data manipulation that could affect the result. In those cases, I had to take a leap and guess and do some trial and error to get close to the final intended result, which of course created uncertainty in the process.

Another challenge could be the obvious differences between RStudio and Stata. For example, how to deal with blanks or missing values, or with errors, or even rounding decimals for some rows or variables, but for other not, can sometimes lead to subtle but real mismatches. These differences are mostly minor, sometimes it's just a tenth of a percentage point, but they're a reminder of how much software defaults can quietly shape results.

Recreating the exact look of the original figures also turned out to be trickier than expected. ggplot2 gives you tons of control, but its defaults are miles apart from Stata's. I found myself constantly adjusting axis spacing, legends, and font sizes, and even then, it wasn't always a perfect match. At some point, I had to make a call between staying visually identical and making the plot easier to read, and I leaned toward clarity.

9 What I would do with More Time and Resources

If I had more time, or maybe someone to collaborate with, I would have been interested into taking the replication process to a few new directions:

Audit each step: I'd do a more rigorous comparison of outputs between R and Stata by checking intermediate datasets line by line or using checksums. That would help catch small differences early and make it easier to trace where they come from.

Modularize the code: Right now, the code works, but it's not very portable. With more time, I'd break it into reusable functions, document each step, and maybe even add some light testing. It'd make it much easier to maintain or adapt down the road.

Automated plot checking: I would look into something like vdiffr to automate visual comparisons. That would be helpful for finding any layout or styling mismatches without having to eyeball every graph and table manually.

I could also focus on assimilating the data behind the code and reproducing it, not only by replicating a graph, but also by replicating a whole paper section.

Expand the analysis to another industry: Given more time I would have tried to adapt the code to pricing data from another industry or context. It would be interesting to see if the patterns the paper identifies still apply, or if their findings are perhaps more platform-specific than what they really seem.

10 Conclusions and Reflections on the Process

What this project really made me not only understand, but also value, was how non – linear replications and coding can be, even when you have a well-organized, detailed and thorough replication package. Translating someone else's thinking process, ideas and of course, workflow, from one programming language to another isn't just about being careful. It's about understanding the concept, intent, navigating its processes, and deciding what is essential vs. what is just a fancy layout.

Furthermore, I think that another thing that stood out throughout the replication process was how much you can learn from coding, and from a paper itself by trying to replicate an author's work. Going through ambiguous parts of the code forced me to engage more deeply with the logic behind each decision, whether that meant asking myself or wondering why certain filters were applied, or what behavior a particular summary statistic is trying to represent.

Even when outputs matched, the process helped me build a more intuitive grasp of the sense, logic and thinking process behind a coding project. In a way, replicating the paper wasn't just about reproducing results, it was about reconstructing the reasoning behind them with the help of a new programming language: R.

It also made me wonder what can count as a "successful" replication process. I could have bent over backward to make R behave exactly like Stata, but that was not really the point the point was learning how to code and replicate the authors' intent by having a detailed thinking process. Instead, I focused on replicating the underlying logic and results, even if the presentation or structure looked a little different. And honestly, that felt like a more honest and real approach.

"Successful replication isn't about looking identical. It's about reaching the same conclusion through sound logic and thoughtful reconstruction."

10.1 Potential Further Investigations and Readings

- **Extend to other industries:** Re-run the pipeline using pricing data from grocery, electronics, or hospitality sectors to see whether algorithmic pricing behaves similarly.

- **Explore collusion dynamics:** Build on papers like the one written by Calvano et al. (2020) on tacit collusion through pricing bots to test whether similar mechanisms surface in our replicated dataset.

- **Causal inference:** Leverage methods like difference-in-differences or synthetic control to assess the causal effect of algorithmic pricing tools on market price levels.

- **More advanced replication tools:** Read about vdiffr (visual regression testing in R) and testthat for enhancing reproducibility through automated checks of tables and plots.

11 Appendix.

11.1 Links of interest.

1. Landing page
2. Presentation
3. Project Repository

11.2 Graph Code Chunk

```r
#|
library(haven)
library(dplyr)
library(ggplot2)
library(janitor)
library(patchwork)
# Load and clean
data <- read_dta("C:/Users/Christian Casas/OneDrive - studhsf/Documents/Masters - HS Fresenius/2nd Semester
  janitor::clean_names()

# Filter for Xyzal 80ct Tablet (non-multipack)
data_xyzal <- data %>%
  filter(
    brand == "Xyzal",
    form == "Tablet",
    size == 80,
    multipack == 1,
    flag_imputed_price != 1
  ) %>%
  distinct(website, period_id, .keep_all = TRUE)

# Get upper Y-axis limit for clean breaks
x_max <- max(ceiling(max(data_xyzal$price, na.rm = TRUE) / 200) * 200, 12000)


# Filter for Claritin
data_claritin <- data %>%
  filter(
    brand == "Claritin",
    form == "Tablet",
    size == 70,
    multipack == 1,
    flag_imputed_price != 1
  ) %>%
  distinct(website, period_id, .keep_all = TRUE)

x_max <- max(ceiling(max(data_claritin$price, na.rm = TRUE) / 200) * 200, 12000)

#1
p1 <- ggplot(data_xyzal, aes(x = period_id, y = price, color = website)) +
  geom_line(size = 0.9) +
  scale_color_manual(values = c("A" = "black", "B" = "#98bf64", "C" = "darkblue", "D" = "brown", "E" = "#DA
  scale_x_continuous(limits = c(0, x_max), breaks = seq(0, x_max, by = 2400)) +
  labs(title = "Panel A. Xyzal, tablets, 80 count", x = "Hours Elapsed in Sample", y = "Price") +
  theme_minimal(base_size = 14) +
  theme(legend.position = "bottom", legend.title = element_blank())
# First plot
p2 <- ggplot(data_claritin, aes(x = period_id, y = price, color = website)) +
  geom_line(size = 0.9) +
  scale_color_manual(values = c("A" = "black", "B" = "#98bf64", "C" = "darkblue", "D" = "brown", "E" = "#DA
  scale_x_continuous(limits = c(0, x_max), breaks = seq(0, x_max, by = 2400)) +
  labs(title = "Panel B. Claritin, tablets, 70 count", x = "Hours Elapsed in Sample", y = "Price") +
  theme_minimal(base_size = 14) +
  theme(legend.position = "bottom", legend.title = element_blank())

# Display plots side by side
p1 + p2 + plot_layout(ncol = 2)
```

11.3 Table code chunk

```
library(haven)
library(dplyr)
library(tidyr)
library(janitor)
library(gt)
library(tibble)

# Load and clean
df <- read_dta("C:/Users/Christian Casas/OneDrive - studhsf/Documents/Masters - HS Fresenius/2nd Semester/D
  clean_names() %>%
  filter(!is.na(price), flag_imputed_price != 1)


# Ensure proper types
df <- df %>%
  mutate(
    date = as.Date(date),
    price_change = as.integer(price_change),
    is_observed = as.integer(is_observed)
  )

# Construct abs_price_change only where price_change occurred
df <- df %>%
  group_by(website, product_website_id) %>%
  arrange(date, .by_group = TRUE) %>%
  mutate(abs_price_change = if_else(price_change == 1, abs(price - lag(price)), NA_real_)) %>%
  ungroup()

# Collapse to daily product-level data
daily_df <- df %>%
  group_by(website, product_website_id, date) %>%
  summarise(
    n_price_change = sum(price_change, na.rm = TRUE),
    abs_price_change = sum(abs_price_change, na.rm = TRUE),
    observations = sum(is_observed, na.rm = TRUE),
    has_price_change = as.integer(any(price_change == 1)),
    price = mean(price, na.rm = TRUE),
    .groups = "drop"
  )

# Collapse to website-date level
summary_df <- daily_df %>%
  group_by(website, date) %>%
  summarise(
    n_products = n(),
    n_price_change = sum(n_price_change, na.rm = TRUE),
    abs_price_change = sum(abs_price_change, na.rm = TRUE),
    has_price_change = sum(has_price_change, na.rm = TRUE),
    observations = sum(observations, na.rm = TRUE),
    price_mean = mean(price, na.rm = TRUE),
    price_sd = sd(price, na.rm = TRUE),
    price_10 = quantile(price, 0.10, na.rm = TRUE),
    price_90 = quantile(price, 0.90, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  mutate(
    price_change_per_product = n_price_change / n_products,
    has_price_change_per_product = has_price_change / n_products,
    obs_per_product = observations / n_products,
    avg_abs_price_change = abs_price_change / n_price_change
  )

# Compute final summary stats per website
stats_by_website <- summary_df %>%
  group_by(website) %>%
  summarise(
    `Count of Products` = mean(n_products, na.rm = TRUE),
    `Observations per Product` = mean(obs_per_product, na.rm = TRUE),
    `Price: Mean` = mean(price_mean, na.rm = TRUE),
    `Price: 10th Percentile` = mean(price_10, na.rm = TRUE),
    `Price: 90th Percentile` = mean(price_90, na.rm = TRUE),
    `Mean Absolute Price Change` = mean(avg_abs_price_change, na.rm = TRUE),
    `Price Changes per Product` = mean(price_change_per_product, na.rm = TRUE),
    `Share of Products with a Price Change` = mean(has_price_change_per_product, na.rm = TRUE),
    .groups = "drop"
  )

# Add "Total" row across all websites
total_row <- summary_df %>%
  summarise(
    website = "Total",
    `Count of Products` = mean(n_products, na.rm = TRUE),
    `Observations per Product` = mean(obs_per_product, na.rm = TRUE),
    `Price: Mean` = mean(price_mean, na.rm = TRUE),
    `Price: 10th Percentile` = mean(price_10, na.rm = TRUE),
    `Price: 90th Percentile` = mean(price_90, na.rm = TRUE),
    `Mean Absolute Price Change` = mean(avg_abs_price_change, na.rm = TRUE),
    `Price Changes per Product` = mean(price_change_per_product, na.rm = TRUE),
    `Share of Products with a Price Change` = mean(has_price_change_per_product, na.rm = TRUE)
  )

# Combine
final_stats <- bind_rows(stats_by_website, total_row)

# Round for display
final_stats_rounded <- final_stats %>%
  mutate(across(where(is.numeric), ~ round(.x, 2))) %>%

  column_to_rownames("website") %>%
  t() %>%
  as.data.frame()%>%
  rownames_to_column("Statistic")


# Display
final_stats_rounded %>%
  gt() %>%
  tab_header(title = "Table 1 - Daily Statistics for Hourly Price Data") %>%
  opt_table_font(size = 8) %>%
  tab_options(
    table.width = pct(60),
    column_labels.font.size = "smaller",
    data_row.padding = px(2)
  )
```

12 Affidavit

I hereby affirm that this submitted paper was authored unaided and solely by me. Addi-tionally, 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 author-ship. 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, email, matriculation number).
[x] The report contains an abstract.
[x] The presentation and the report contain a bibliography, created using BibTeX with APA citation style.
[x] The report contains R code that proofs the students) expertise in coding.
[x] The report includes an introduction to guide the reader and a conclusion summarizing the work and discussing potential further investigations and readings, respectively.

[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, as detailed here: https://github.com/hubchev/make_a_pull_request.
[x] The link to the presentation and the handout published on GitHub.

Christian Faruk Rodriguez Casas, 18/07/2025, Cologne, Germany

13 References

Board, S., & Meyer-ter-Vehn, M. (2022). A reputational theory of firm dynamics. *American Economic Journal: Microeconomics*, *14*(2), 44–80. https://doi.org/10.1257/mic.20190376

Brown, Z. Y., & MacKay, A. (2023). Competition in pricing algorithms. *American Economic Journal: Microeconomics*, *15*(2), 109–156. https://doi.org/10.1257/mic.20210158

Calvano, E., Calzolari, G., Denicolò, V., & Pastorello, S. (2020). Artificial intelligence, algorithmic pricing, and collusion. *American Economic Review*, *110*(10), 3267–3297. https://doi.org/10.1257/aer.20190623

Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org

Wickham, H., François, R., Henry, L., Müller, K., & Vaughan, D. (2025). *Dplyr: A grammar of data manipulation*. https://dplyr.tidyverse.org

Wickham, H., Miller, E., & Smith, D. (2025). *Haven: Import and export 'SPSS', 'stata' and 'SAS' files*. https://haven.tidyverse.org