Data Replication

Tony Osei

Introduction

- Objective: Reproduce the graphs using RStudio. These graphs show "Heterogeneity in Pricing Technology by Hour of the Week"
- Tools used: Data from `analysis_data.dta` , `data.table`, `ggplot2`, and `lubridate` packages
- Outcome: Understand retailer pricing behavior by hour and day of the week.
- In this paper, Brown & MacKay (2023), show that pricing behavior varies significantly across retailers.
- We used `ggplot2` for plotting (Wickham (2016)) and built this deck with Quarto (Posit, PBC (2022)).

Replication Code

```
library(haven)
library(data.table)
library(dplyr)
library(ggplot2)


data <- read_dta("C:/Users/attef/OneDrive/Documents/Replicaproject/Replica/analysis/data/anal
df <- as.data.table(haven::read_dta("C:/Users/attef/OneDrive/Documents/Replicaproject/Replica


# Filter for Retailer A
retailer_A <- df_hourly[website == "A"]

ggplot(retailer_A, aes(x = hourofweek, y = hourly_dist)) +
  geom_line(color = "black", linewidth = 1) +
```

```r
  # X-axis: tick every 24 hours (no labels)
  scale_x_continuous(
    breaks = seq(0, 168, by = 24),
    limits = c(0, 168),
    expand = c(0, 0)
  ) +

  # Y-axis: 0% to 1% for Retailer A
  scale_y_continuous(
    limits = c(0, 1),
    breaks = seq(0, 1, by = 0.2),
    labels = function(x) sprintf("%.1f", x)
  ) +

  # Vertical dashed lines at day boundaries
  geom_vline(xintercept = seq(24, 144, by = 24), linetype = "dashed", color = "gray60") +

  # Add day labels as text (not tick labels)
  annotate("text", x = 12, y = 0, label = "Sat", vjust = 1.5, size = 4) +
  annotate("text", x = 36, y = 0, label = "Sun", vjust = 1.5, size = 4) +
  annotate("text", x = 60, y = 0, label = "Mon", vjust = 1.5, size = 4) +
  annotate("text", x = 84, y = 0, label = "Tue", vjust = 1.5, size = 4) +
  annotate("text", x = 108, y = 0, label = "Wed", vjust = 1.5, size = 4) +
  annotate("text", x = 132, y = 0, label = "Thu", vjust = 1.5, size = 4) +
  annotate("text", x = 156, y = 0, label = "Fri", vjust = 1.5, size = 4) +

  labs(
    title = "Panel A. Retailer A",
    x = "Hour of Week",
    y = "Percent of Price Changes"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    panel.grid.minor = element_blank(),
    axis.text.x = element_blank(),   # Hide tick labels
    axis.ticks.x = element_blank(),
    plot.title = element_text(hjust = 0.5)
  )


ggplot(df_hourly[website == "B"], aes(x = hourofweek, y = hourly_dist)) +
  geom_line(color = "black", linewidth = 1) +
```

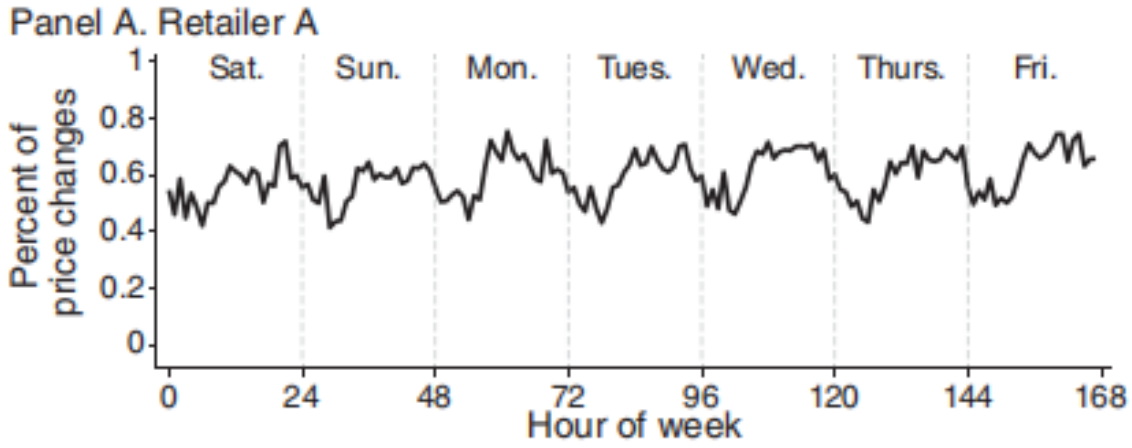
```
  scale_x_continuous(
    breaks = seq(0, 168, by = 24),
    limits = c(0, 168),
    expand = c(0, 0)
  ) +
  scale_y_continuous(
    limits = c(0, 1),
    breaks = seq(0, 1, by = 0.2),
    labels = function(x) sprintf("%.1f", x)
  ) +
  geom_vline(xintercept = seq(24, 144, by = 24), linetype = "dashed", color = "gray60") +
  annotate("text", x = seq(12, 156, by = 24), y = 0, label = c("Sat", "Sun", "Mon", "Tue", "
  labs(
    title = "Panel B. Retailer B",
    x = "Hour of Week",
    y = "Percent of Price Changes"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    axis.text.y = element_text(size = 10),
    plot.title = element_text(hjust = 0.5),
  )


ggplot(df_hourly[website == "C"], aes(x = hourofweek, y = hourly_dist)) +
  geom_line(color = "black", linewidth = 1) +
  scale_x_continuous(
    breaks = seq(0, 168, by = 24),
    limits = c(0, 168),
    expand = c(0, 0)
  ) +
  scale_y_continuous(
    limits = c(0, 8),
    breaks = seq(0, 8, by = 2),
    labels = function(x) sprintf("%.0f", x)
  ) +
  geom_vline(xintercept = seq(24, 144, by = 24), linetype = "dashed", color = "gray60") +
  annotate("text", x = seq(12, 156, by = 24), y = 0, label = c("Sat", "Sun", "Mon", "Tue", "
  labs(
    title = "Panel C. Retailer C",
```

```
    x = "Hour of Week",
    y = "Percent of Price Changes"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    axis.text.y = element_text(size = 10),
    plot.title = element_text(hjust = 0.5),
  )

ggplot(df_hourly[website == "D"], aes(x = hourofweek, y = hourly_dist)) +
  geom_line(color = "black", linewidth = 1) +
  scale_x_continuous(
    breaks = seq(0, 168, by = 24),
    limits = c(0, 168),
    expand = c(0, 0)
  ) +
  scale_y_continuous(
    limits = c(0, 25),
    breaks = seq(0, 25, by = 5),
    labels = function(x) sprintf("%.0f", x)
  ) +
  geom_vline(xintercept = seq(24, 144, by = 24), linetype = "dashed", color = "gray60") +
  annotate("text", x = seq(12, 156, by = 24), y = 0, label = c("Sat", "Sun", "Mon", "Tue", "
  labs(
    title = "Panel D. Retailer D",
    x = "Hour of Week",
    y = "Percent of Price Changes"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    axis.text.y = element_text(size = 10),
    plot.title = element_text(hjust = 0.5),
  )


ggplot(df_hourly[website == "E"], aes(x = hourofweek, y = hourly_dist)) +
  geom_line(color = "black", linewidth = 1) +
  scale_x_continuous(
```

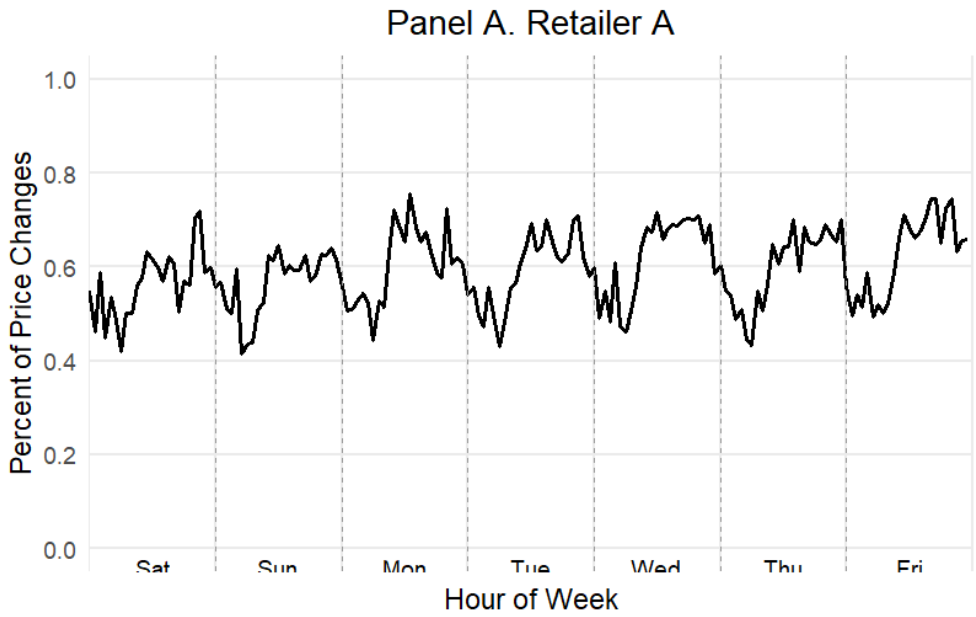
```
  breaks = seq(0, 168, by = 24),
  limits = c(0, 168),
  expand = c(0, 0)
) +
scale_y_continuous(
  limits = c(0, 60),
  breaks = seq(0, 60, by = 10),
  labels = function(x) sprintf("%.0f", x)
) +
geom_vline(xintercept = seq(24, 144, by = 24), linetype = "dashed", color = "gray60") +
annotate("text", x = seq(12, 156, by = 24), y = 0, label = c("Sat", "Sun", "Mon", "Tue", "
labs(
  title = "Panel E. Retailer E",
  x = "Hour of Week",
  y = "Percent of Price Changes"
) +
theme_minimal(base_size = 14) +
theme(
  axis.text.x = element_blank(),
  axis.ticks.x = element_blank(),
  axis.text.y = element_text(size = 10),
  plot.title = element_text(hjust = 0.5),
)
```

First Graph

Original

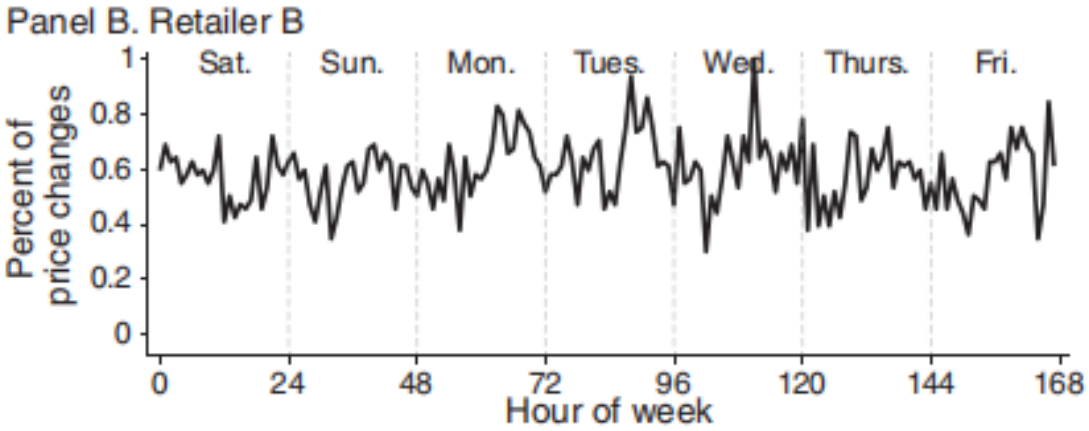


Replication

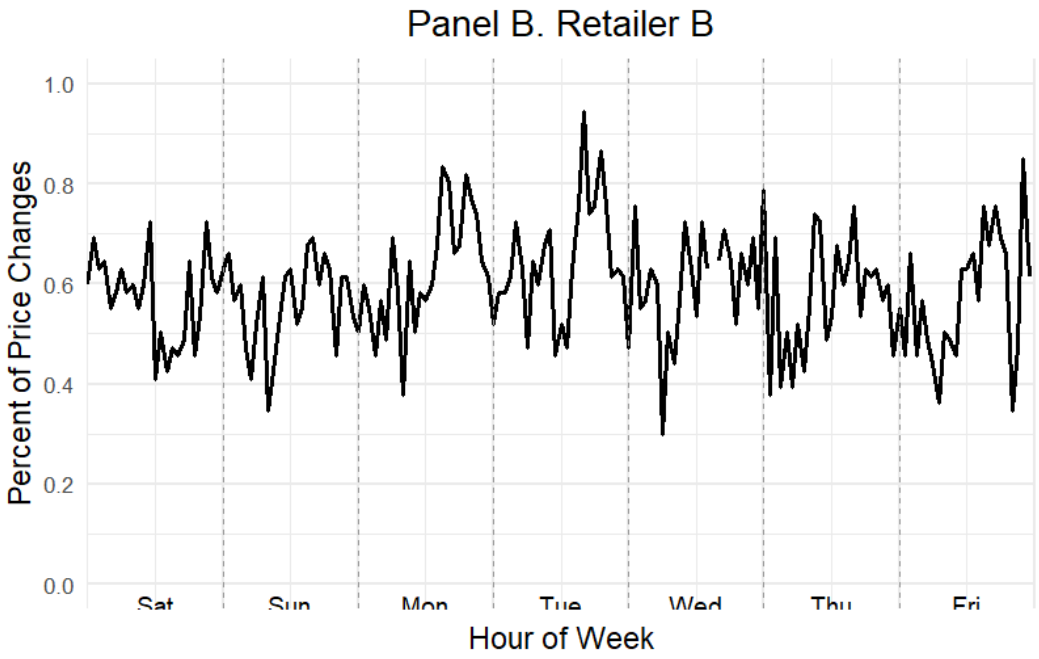


Second Graph

Original

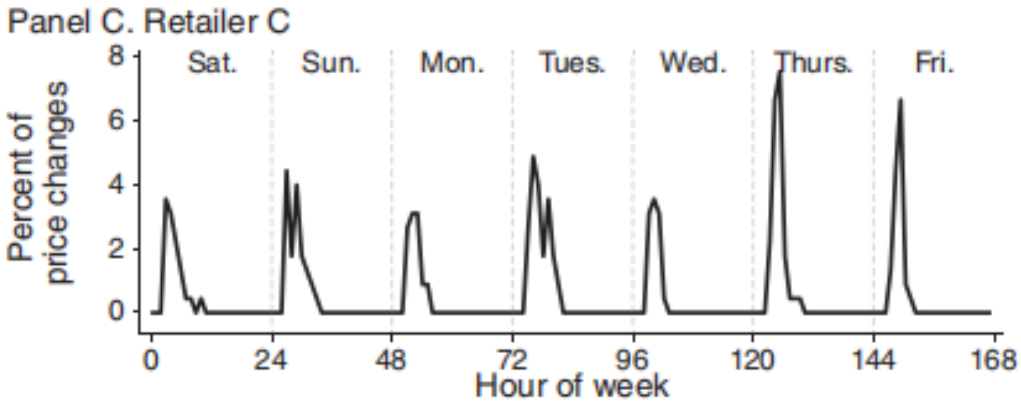


Replication

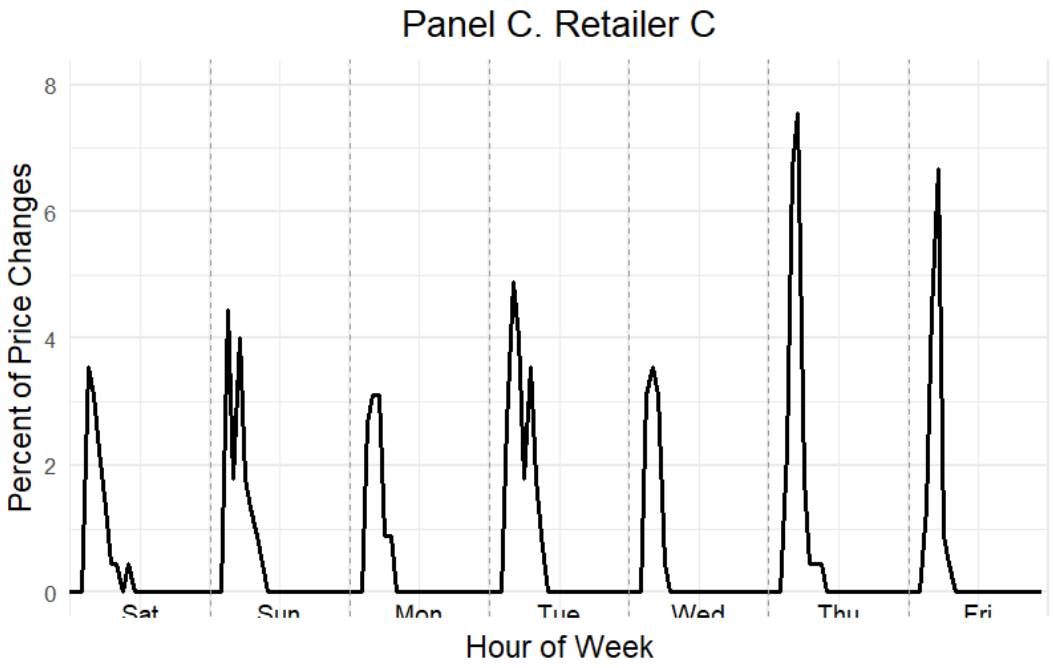


Third Graph

Original

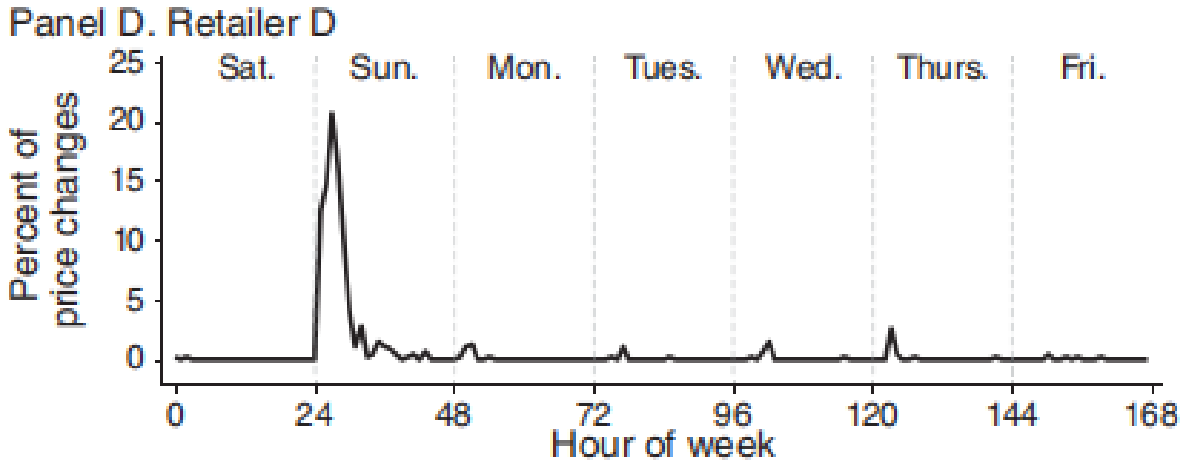


Replication

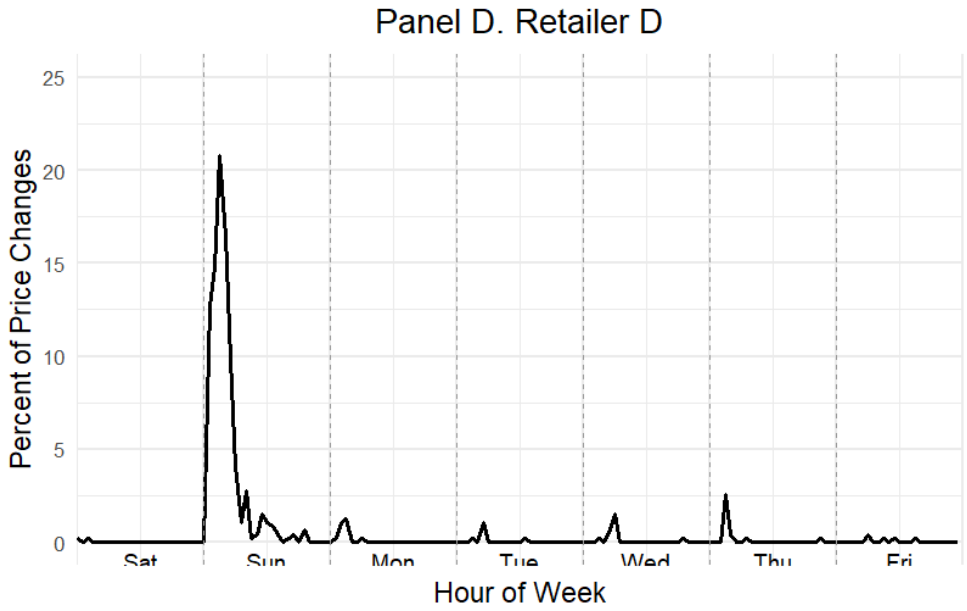


Fourth Graph

Original

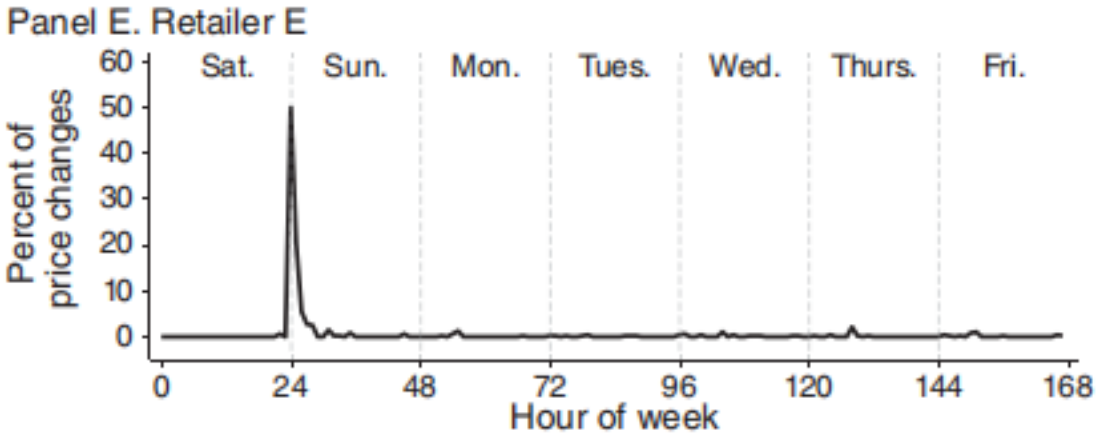


Replication

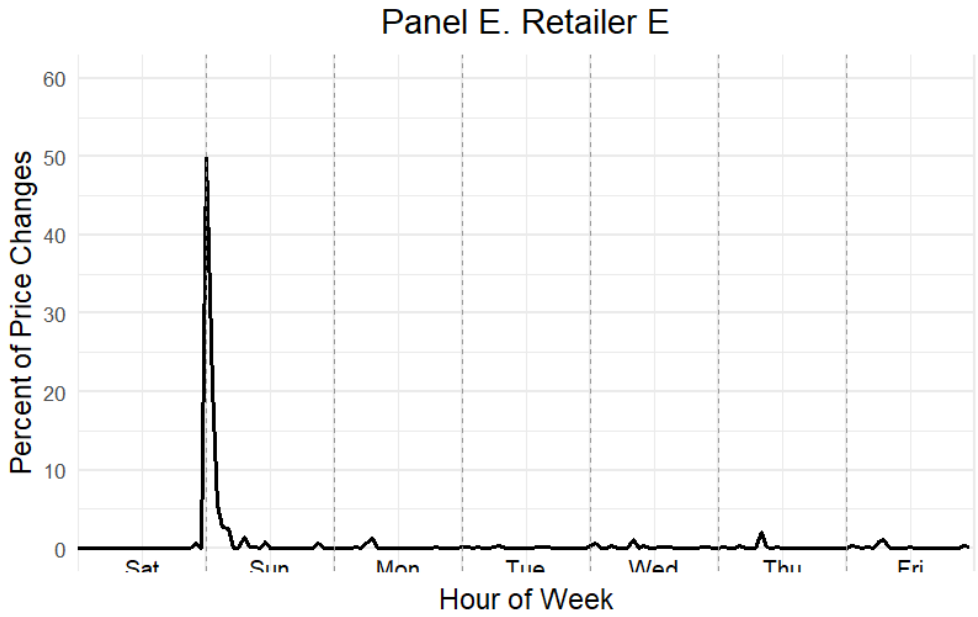


Fifth Graph

Original



Replication



Problems

1. Pushing to GitHub Rejected Due to Large File Size

Solution:

Created a `.gitignore` file to exclude the large data file from being tracked.

Used `git rm --cached` to untrack the file.

Attempted to clean Git history using BFG Repo-Cleaner, which required downloading and setting up **Java** on the system to run `.jar` files.

Once cleaned, the project was pushed successfully to GitHub.

2. Missing or Misaligned axis

Problems

3. Missing packages or functions in R.
4. Data File not found/ Path errors.
5. Quarto reference file not found

Conclusion

- Successfully replicated the *Hour-of-Week Price Change* plots from , Brown & MacKay (2023), albeit challenges with displaying graphs fully.
- Addressed challenges with data formatting, visualization, and GitHub publishing
- Leveraged R, Quarto, and ggplot2 for reproducible and shareable analysis

Looking ahead: - Extend this work to other figures in the paper - Explore dynamic or interactive presentations using Quarto + Shiny

References

Brown, Z. Y., & MacKay, A. (2023). Competition in pricing algorithms. *The Quarterly Journal of Economics*, *138*(1), 113–157. https://doi.org/10.1093/qje/qjac035

Posit, PBC. (2022). *Quarto: Scientific and technical publishing system*. https://quarto.org

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