Online Appendix for "Inventory Information Frictions Explain Price Rigidity in Perishable Groceries "*

Naveed Chehrazi[†], Robert Evan Sanders[‡], Yannis Stamatopoulos[§]

July 3, 2024

Abstract

In this document, we provide a number of additional exhibits in the order they are referenced in the main text and in appendices.

keywords: price rigidity, revenue management, information frictions, perishable goods

---

[*] Formerly titled: "Inventory Record Inaccuracy Explains Price Rigidity in Perishable Groceries." We thank the seminar participants at Northwestern University, London Business School, Indiana University, University of Pennsylvania, Yale University, Boston University, University of Virginia, and Amazon Science. We also thank the participants at Harvard's 2021 Consortium for Operational Excellence in Retailing (COER). Finally, we thank Kathleen Li, Kanishka Misra, Christopher Ryan, Steve Gilbert, Diwakar Gupta, and Lingxiu Dong for their feedback on the manuscript.

[†] Washington University in Saint Louis, Olin Business School.

[‡] The University of California, San Diego, Rady School of Management.

[§] The University of Texas at Austin, McCombs School of Business.

1 Additional Exhibits

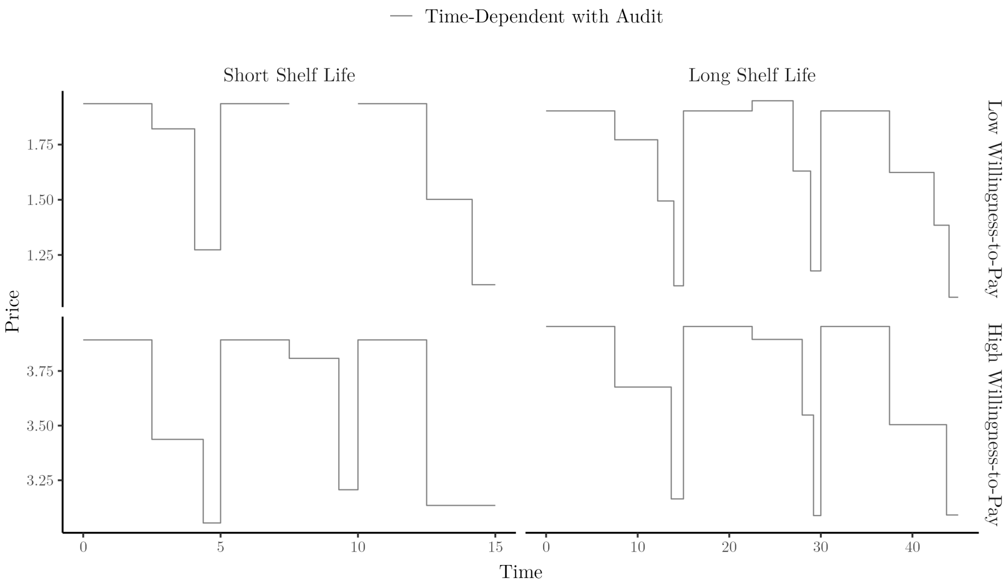

Figure O1:

This figure demonstrates "typical" time-dependent price paths for different primitives with menu costs and inventory audits. To generate these graphs, we use the same parameters and approach as we do in Figure 2, except that we allow for a free mid-cycle inventory audit and a free price update immediately after.

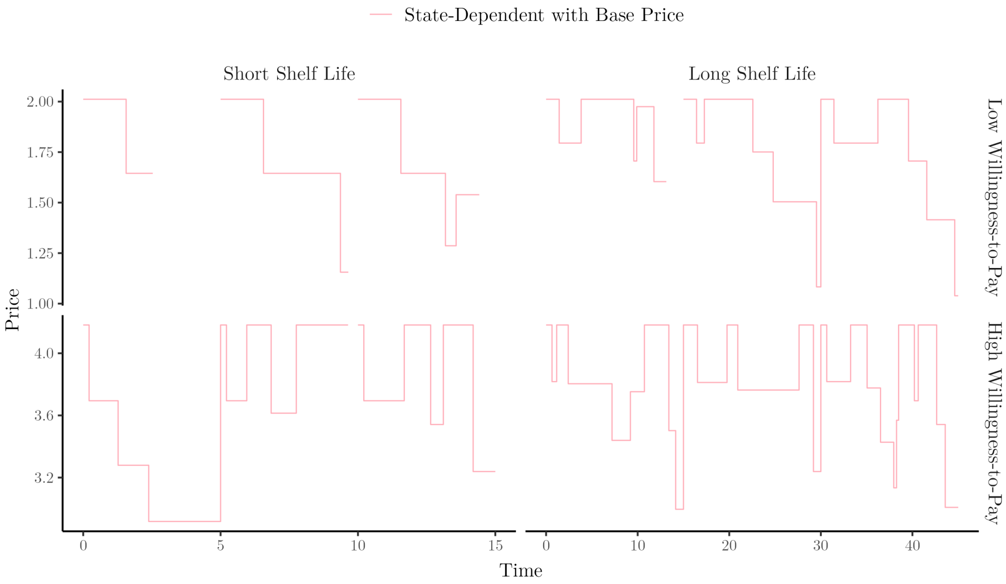

Figure O2:

This figure demonstrates "typical" state-dependent price paths for different primitives with menu costs and a base-price contraint. To generate these graphs, we use the same parameters and approach as we do in Figure 2, except that the price path is bounded above by $1.1 \times_{p \geqslant 0} (p-c)G(p) = 1.1 \times (c+\gamma)$.

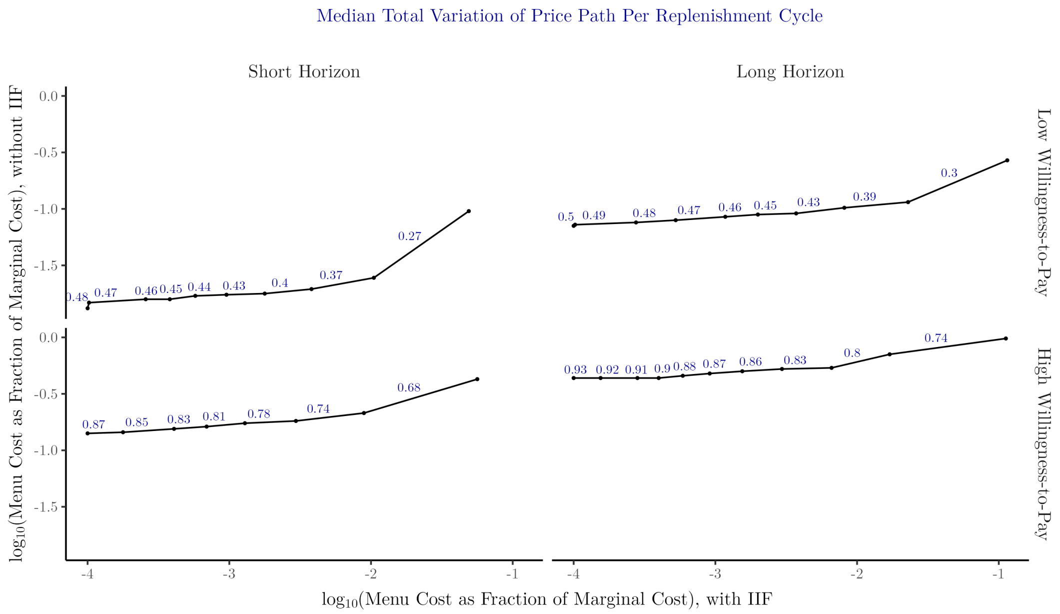

Figure O3:

This figure illustrates the menu costs that are required to rationalize different price rigidity-levels with IIF (i.e., under time-dependent pricing) and without IIF (i.e., under state-dependent pricing) when price rigidity is measured by total variation (defined as the sum of absolute changes in a price trajectory). To generate these graphs, we use the same parameters and approach as in Figure 3. Within each of the four panels, the projection of the line segment labeled by a total variation on the horizontal axis corresponds to the range of menu costs that generates the stated total variation as the median outcome under (optimal) time-dependent pricing. The projection of the same line segment on the vertical axis corresponds to the analogous range of menu costs under (optimal) state-dependent pricing. For example, in the long-horizon, high-willingness-to-pay case, a menu cost equal to 1% of marginal cost (-2 on the horizontal axis) generates total variation of 0.8 with IIF. But to generate the same total variation without IIF, the model would need a menu cost equal to 50% of marginal cost (-0.3 on the vertical axis).

2 Empirical Details

2.1 U.K. pilot

The U.K.-pilot data come from a store that installed ESLs for all of its products. The store is part of an international retail chain headquartered in continental Europe. We observe transaction-line data containing identical fields before and after the installation of ESLs. Our raw data contain 688,334 transactions across 13,865 products and 99 categories. We make the following transformations to construct our final sample:

1. Rarely, two transaction lines in the same basket for the same product have incongruent promotion statuses. This error occurs in 672 transactions. We drop these transactions. One product was also dropped.

2. Rarely, two transaction lines in the same basket for the same product have incongruent promotion prices. This error occurs in 152 transactions. We drop these transactions.

3. Some products have a placeholder product beginning with "009999999999998." We cannot tell such products apart. In total, there are 19 of these specific products. We drop all such products, which removes 14,629 transactions. After further inspection, we find 88 more products that seem to be

placeholders (e.g., products beginning with “00000000000”). We drop these products, which removes 11,976 transactions.

4. We drop the technology-installation window (January 1, 2015 – March 3, 2015) and balance the sample on both sides (as explained in the main text). We drop these time periods, which removes 417,322 transactions and 5,593 products. Most of the dropped transactions occurred in the latter part of the post-installation window, i.e., more than 146 days after March 4, 2015. Similarly, most of the products dropped were only sold during in the latter part of the post-installation window.

5. Because we infer prices from transaction data, we require enough transactions to credibly measure price rigidity. We keep products that have at least 20 transactions in both the pre- and post-installation period, which removes 6,582 products and 50,252 transactions.

6. For the same reasons as in (5), we keep only the products that have at least one sale in at least 16 weeks in both the pre- and the post-installation period, which removes 642 products and 26,093 transactions.

After applying these filters, we construct a minute-by-minute panel of prices for each product, starting from the product’s first transaction line and ending at its last.[1]

We construct the price panel as follows:

1. When we observe a zero-sales time window that is more than ten days long for a particular product, we assume that the product is out of stock and drop that time window for that product.

2. For zero-sales time windows that are fewer than ten days long, we impute the product’s price backward between transactions (i.e., in a “right-continuous” manner).

3. After applying the promotion fixes described above, we prune no further prices.

2.2 E.U. pilot

The E.U.-pilot data come from a store that installed ESLs and expanded barcodes for 26 meat and poultry products. The store is part of a European grocery chain, and the products are supplied by a cooperating wholesaler. The raw pre-installation data contain 331,675 transactions across 641 various meat and poultry products. The first week post installation was dedicated to quality assurance, so we drop this week from the data. After retaining only our focal 23 products and dropping returns, which make up 0.02% of transaction lines, there are 19,288 transactions in the pre-installation period and 30,579 transactions in the post-installation period.

After applying the previous filters, we construct a minute-by-minute panel of prices for each product, starting from the product’s first transaction line and ending at its last.[2] Unlike in the U.K. pilot, in the E.U. store’s pre-installation data, we do not observe indicators for promotions nor do we observe manual price overrides (e.g., flags for waste discounts). Based on interviews with the technology provider, the retailer only infrequently offered clearance sales based on an item’s expiration date in the pre-installation period, and the sales were implemented with a sticker offering 35–50% off the shelf price.

---

[1] In our U.K. store’s data, there are only 115 date–minute–products for which there is more than one unique price paid. For such instances, we keep the maximum observed price.

[2] In our E.U. store’s data, during the pre-intervention period, there are only 11 date–minute–products for which there is more than one unique price paid. In the post-intervention period, there are only 405 date–minute–products for which there is more than one unique price paid. For such instances, we keep the maximum observed price.

In the post-installation period, we observe inventory-based price indicators as well as promotion indicators. We also observe shelf prices in a separate data base. When we construct our price panel, we impute prices from the transaction line data to be consistent with the pre-installation data and the U.K. pilot. Our results are robust to either using the full shelf-price information or inferring shelf prices from transaction data.

We construct the price panel as follows:

1. When we observe a zero-sales time window that is more than ten days long for a particular product, we assume that the product was out of stock and drop that time window for that product.

2. For zero-sales time windows that are fewer than ten days long, we impute the product's price backward between transactions in a "right-continuous" manner.

3. In the pre-installation period, we observe several prices that are almost certainly errors. These prices are almost universally at unusual values (e.g., €0.42, €0.02, etc.) and last for only a single transaction. We drop such prices unless they are between 35% and 50% lower than the median weekly price, in which case we retain them and label them clearance sales. Keeping these prices risks including some errors (incorrectly classified as clearance sales), but their inclusion biases the analysis against our conclusions.

3 Post-installation Prices in the E.U. Store are Inventory-based

Our post-installation data contain a flag indicating when a price change was inventory-based. This flag demonstrates that the majority of the E.U. store's post-installation price updates condition on inventory records: across products, the median fraction of inventory-based price updates is 68.4%, the 25th percentile is 65.3%, and the 75th percentile is 78.3%.

Nevertheless, for completeness, we also quickly verify that inventory information—provided to the store by the expanded barcodes—does indeed predict the E.U. store's price-setting behavior post-installation. Specifically, for each of the E.U. store's 23 products, we predict two price-setting outcomes: whether the store changes the product's price, and the product's new price after a price change. For each outcome, we use two sets of features: demand features, and demand + inventory features. If, for both price-setting outcomes and for most products, the model's predictive power substantially increases when adding inventory features to detailed demand features, then we can safely say that the store frequently engages in inventory-based pricing.

3.0.1 Pricing-decision panels.

After the installation of the ESLs + expanded barcodes, the E.U. store reviews prices at pre-determined times throughout each day. We construct a panel of pricing decisions for each product using the store's shelf-price data. Specifically, in the raw post-installation data, we observe 70,047 price reviews and 6,386 changes across the 23 products. We apply the following filters to construct our final sample:

1. We drop the 138 price reviews that did not occur at one of the pre-determined times.

2. We drop the 31 price changes that occur between regular price reviews.

3. We drop the first twenty days of each product's panel to be able to compute lagged demand features.

Applying these filters leaves us with 64,674 price reviews and 6,056 price changes.

3.0.2 Features.

Our (product-level) demand features are: day of the week; hour of the day; a pre-planned promotion flag; the product's average sales per hour between each of the last ten price reviews; the minimum shelf price for each of the last ten price reviews; the base price of the product 5, 10, 15, and 20 days ago; and a flag for whether a competing product's price—outside the focal 23 products—changed recently. Our (product-level) inventory features are counts for items that expire in τ days, where τ runs from 1 to the product's maximum shelf-life. Note that most of our demand features indirectly include inventory information in them. For example, a high recent sales rate might imply presently low inventories, or a recent clearance sale (reflected in the minimum shelf price features) could indicate a product near expiration. Therefore, our comparison (between models that use demand and demand + inventory features) almost certainly understates the incremental predictive power of inventory information in predicting the store's price-setting behavior.

3.0.3 Model Selection.

We use our panel to train and test our predictive models for each price-setting outcome and feature set. For our models, we use gradient-boosted trees (XGBoost) with a quadratic loss function and an 80/20 split for training/testing. We set the learning rate equal to 0.1, use 80% subsampling to train each learner, and select our maximum tree depth and minimum child weight from the grid $\{1, 2, ..., 210\} \times \{2, ..., 9\}$. Specifically, for each point in this grid, we find the number of weak learners in our ensemble that minimizes the out-of-sample mean absolute error (MAE) in a 5-fold cross-validation. Then, we choose the grid point that achieves the global minimum MAE and use the corresponding number of weak learners. We use out-of-sample R^2 to evaluate the model's predictive performance. We apply the preceding steps for 50 different (random) 80/20 splits for each product.

3.0.4 Results.

When predicting whether prices change, we get a median (across product–80/20 splits) out-of-sample R^2 of 34.9% with demand features alone and a median out-of-sample R^2 of 55.2% with demand + inventory features (Figure O4). A Kolmogorov-Smirnov test rejects the null hypothesis that the out-of-sample R^2 distributions generated with and without inventory features are the same, at the $p < 10^{-15}$ level. In fact, adding inventory features increases the out-of-sample R^2 for 99.5% of the product–80/20 splits. To assess the economic significance of this increase in predictive power, we predict pricing outcomes with a third set of features. Specifically, we calculate the average hourly sales rate between the current and next price reviews and add it to the demand features. Note that this additional feature contains information from the future; that is, it contains information that could be a *result* of the decision made in the current price review. For example, a higher future sales rate could be indicative of a discount introduction in the current price review. Similarly, a lower future sales rate could be indicative of a discount removal in the current price review. The median out-of-sample R^2 with demand + future demand features is 78.8%. In other words, inventory features close 42.4% of the gap in predictive power between demand and demand + future demand features.

When predicting new price levels, we get a median (across product–80/20 splits) out-of-sample R^2 of 51.3% with demand features alone, and a median out-of-sample R^2 of 57% with demand + inventory features. A Kolmogorov-Smirnov test rejects the null hypothesis that the out-of-sample R^2 distributions generated with and without inventory features are the same, at the $p < 10^{-9}$ level. This time, adding inventory features increases the out-of-sample R^2 for 71.7% of the product–80/20 splits.[3] Interestingly, future demand does not help in predicting price levels: a Kolmogorov-Smirnov test fails to reject the null hypothesis that the out-of-sample R^2 distributions generated using demand features with and without future demand are the same ($p = .991$).

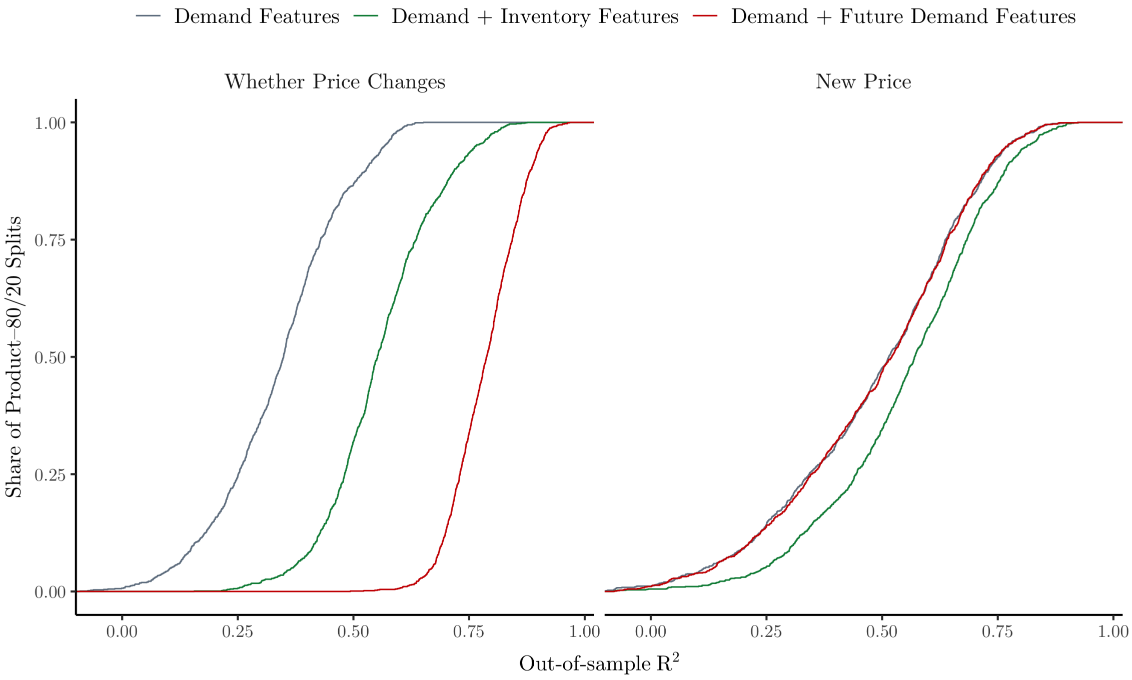

Figure O4:

This figure illustrates that inventory features predict price-setting behavior of the E.U. store post-ESLs + expanded barcodes. To generate these cumulative distribution functions, we first draw 50 random, 80/20 training/testing splits of the data for each of the 23 products. Then, for each product–80/20 split, we: (i) train a model with demand features (i.e., day of the week, hour of the day, pre-planned promotion flag, etc.) and a model with demand + inventory features (i.e., the demand features plus item counts by expiration date) using the training sample, (ii) predict whether a price review leads to a price change (left panel) and the price level conditional on a change (right panel) with each model using the test sample, and (iii) calculate the out-of-sample R^2 associated with these predictions.

---

[3] That adding inventory features to our model improves the out-of-sample R^2 more when predicting price changes than it does when predicting price levels is not too surprising. Inspecting the raw time series of prices, we see a large number of small discounts, occasionally interrupted by large base-price changes. Because they are much larger, these base-price changes disproportionately contribute to the R^2. And these base price changes are not inventory-driven.