A Online Appendix

A.1 Data

A.1.1 Data Source

The data in the Kilts Center are collected by Nielsen and made available through the Marketing Data Center at the University of Chicago Booth School of Business. Information on availability and access to the data can be found at https://research.chicagobooth.edu/kilts/marketing-databases/nielsen.

A.1.2 Store Selection

In the RMS data, Nielsen provides a basic categorization of stores into five "Channel Codes": Convenience, Food, Drug, Mass Merchandise, and Liquor. Of these, we select Food, Drug, and Mass Merchandise chains since the Convenience and Liquor stores are typically not covered in the Homescan data and thus would not be included in our final sample. In the Homescan data, there are more detailed "Retailer Channel Codes" and each store is assigned to one of 66 mutually exclusive categories such as Department Store, Grocery, Fruit Stand, Sporting Goods, and Warehouse Club. Our starting sample of food stores includes all stores that are categorized as "Food" stores in the RMS data. All food stores selected in the final sample fall into the "Grocery" category in the Homescan channel code categorization[41], all drugstores in the "Drug Store" category, and all Mass Merchandise stores in the "Discounters" category.

Some stores change DMA or FIPS code over the time that they are in the sample. Since Nielsen identifies store by the physical location of the store, this occurs because DMA regions or county lines are redefined over the nine years we observe. In other words, the stores themselves are not changing physical locations. For stores that switch, we use the modal DMA and FIPS code. This does not affect how we aggregate store-level demographics for our main analysis.

A.1.3 Demographics

All demographics are zipcode level data from the 2008-2012 5-year ACS. We aggregate this zipcode level demographics into store-level demographics as explained in Section 2. There are two special cases: (i) for one store with missing median home price data, we impute this value by regressing median home price on the other demographics (income, fraction with a bachelor's degree, race, and fraction of urban area); (ii) three drugstores are only visited by one household each, and these households provide a PO Box zipcode as its zipcode, making it impossible to use our usual procedure, so we use county-level demographics for these three stores.

A.1.4 Competition Measures

We use the Homescan panel data to help us construct a measure of competition based on geodesic distance for the food stores. To compute the location of each store, we use the more detailed location information in the Homescan data.[42] First, we assume that each Homescan household lives at the center of its zipcode. For each of the stores in the Homescan dataset, we use a trip-weighted average of the coordinates of each household in order to arrive at an imputed location

[41] The starting sample of 11,501 Food stores also contains some Discount Stores and Warehouse Clubs, as well as some (likely mislabeled) drugstores.

[42] Recall that the location of the store in the Nielsen publicly available data is only recorded up to the 3-digit zipcode or county.

for the store. For our measure of competition for store s, we then count the number of food stores within various distances (e.g., 5 or 10 km) of store s by geodesic distance, i.e., distance as the crow flies. For each store, we count separately the number of stores in the same chain and stores in different chains.

A.1.5 Product Selection

For our main sample, within a module, we keep all the products (UPCs) which satisfy an availability restriction: pooling across chains, a product must have positive revenue in at least 80% of store-weeks. For this calculation we include only store-weeks with at least $100 of sales across all products.

For the sample of food stores, we build three additional product samples for robustness. First, we select a set of chain-specific store-brand products. (These products are excluded from the main sample.) For each of the 40 modules, we take products with the Nielsen identifier "CTL BR", which identifies store-brand products; we also require that the products have at least 80% availability within their chain. This results in 12,423 generic products in 40 modules. All generic products are used to compute store-level generic product price levels, with an aggregation procedure parallel to the one for the main samples. In our product assortment analysis, the number of generic modules falls to 37 after applying the criteria outlined in A.1.9. These generic products are not comparable across chains.

Second, we identify products in the top decile or bottom decile by yearly revenue. For each product, we compute the annual revenue (taking into account the number of years in the sample) across all food stores. We then identify 135 products in the top 10% of this variable and 135 in the bottom 10%.

Third, we define the module-level baskets. For a given chain we include all products, including generic products, in a module such that the average share of weeks with non-zero sales for that product in that chain is at least 95 percent, where the average is taken across stores. We omit weeks from this calculation in which the store has zero recorded sales in all modules. Since the basket of products is defined by chain, the price indices are not comparable across chains.

A.1.6 Prices

As described in the text, we compute the weekly price P_{sjt} as the ratio of weekly revenue and weekly units sold for that store-product. We apply the following filters: (i) Following the Nielsen manual, we divide the weekly units sold by the variable 'prmult' (price multiplier); (ii) We drop all prices $\leq$ $0.10 since almost surely these represent cases of measurement error. This affects very few observations: 427,353 store-product-weeks (0.02% of observations) in food stores, 170,644 observations (0.07%) in drugstores, and 13,782 observations (0.008%) in mass merchandise stores.

A.1.7 Pairs Dataset for the Analysis of Store Pricing Similarity

For the measure of similarity in pricing across stores, we create a data set of pairs of stores as described in the text. We note the following special cases. For the between-chain pairs, for each chain r, we sample 200 store pairs where one store belongs to chain r and the other store belongs to a different chain r'; we then drop any store pairs where the stores belong to the same *parent_code*. We use the same set of pairs for each product j.

To compute the measure of within-chain similarity cases for the case where stores s and s' are in the same geographic market (DMA), for each chain r, we form 200 new pairs that satisfy the restriction of being in the same DMA, and then proceed as above; similarly for the between-chain similarity. We form new pairs also for the case where stores s and s' are in different geographic markets (DMA) and in different income thirds.

To compute the pricing similarity for pairs of stores within a state, or across state boundaries, we re-draw (up to) 200 pairs for each chain-product satisfying the within-state, and between-state, criteria.

A.1.8 Major Grocer's Data

As additional data, we use the scanner data for 250 stores from a major grocer as in Gopinath et al. (2011). The data-sharing agreement between this retailer and the research community is managed through the SIEPR-Giannini data center (http://are.berkeley.edu/SGDC). Since we want to compare the results using the Nielsen price measure versus the price measures in this major grocer's data, it is important to identify the stores in the Nielsen data which correspond to stores in this additional data set. Since the dataset in Gopinath et al. (2011) covers 2004 to mid-2007, while the RMS data set covers from 2006 on, we focus on the 52 weeks in year 2006. We match the two data sets using the 3-digit zipcode and (with a fuzzy match) using the sum of units sold in 2006 for 10 high-selling products. This results in 132 matches to stores in our main sample, all of which belong to a single Nielsen *retailer_code*. We validate the correctness of the matches using data on price and with an alternative matching algorithm.

For Figure V, we select products that are sold for at least 40 of 52 weeks in 131 of the 132 stores. For the Nielsen dataset, we demean log prices by the Nielsen average price prior to aggregating to the store-level. Since we have exactly one year of data, this process is identical to how we aggregate prices for our benchmark sample. For the major grocer dataset, we demean log prices by the major grocer price average.

A.1.9 Assortment

We construct a store-level assortment price index. We build an index with two characteristics in mind: (i) the index should identify availability of high-price versus low-price products, with the price of a product computed using all stores and chains, not just the store at hand; (ii) we should not conflate variation in price that is due to product size or quantity discounts with variation in product quality. With (ii) in mind, within each of the 40 modules we keep only products with the modal size unit (most commonly ounces) and we divide the products into up to five sub-modules based on product size. Within a sub-module, thus, we are comparing products that are not only the same category, but also of similar size. With (i) and (ii) in mind, we then define for each product j a per-unit constant price $\overline{u}_{jy}$ as the average log price charged for product j in year y across all stores s that carry it, divided by the unit size (e.g., 40 oz). The division by unit size aims to control for residual differences in size within a sub-module. Within a sub-module, we include products in the top 20% by units sold across all food stores (not just the stores in our main sample) to ensure comparability across stores and chains.[43] The assortment price index for a store s, sub-module b and year y is the average per-unit constant price for all the products j in sub-module b with positive sales in year y. To create the final assortment price index for store s, we demean the index by sub-module-year, take the simple average over the years, and then take the simple average over the sub-modules to the store-level.

As additional measures of assortment, within each sub-module, we compute the share of products that are organic, the share of generic products, and the share that is in the top 10% by unit

[43] While there is a possibility that each chain carries unique high and low quality items, we cannot control for chain-level pricing decisions for such products.

price (across all chains). The measures are not meaningful for some sub-modules, e.g., organic batteries. To exclude such cases, for each of these additional measures, we only include submodules where the average of the variable (e.g., share generics) across all stores of all chains is at least 0.01 and the standard deviation is at least 0.005.

A.1.10 Elasticity Computation

For the computation of elasticities, we instrument for log price $logP_{sjt}$ with the log price in the same week t, and the same product j, in other stores of that chain outside the DMA of store s. In cases where a chain operates only in a single DMA, of which there are 14 food chains and 0 in either drug or mass merchandise chains, we split the DMA into two sub-markets and define the instrument using other stores in s's chain located outside s's sub-market. The sub-markets composed of clusters of stores that are defined as follows:

1. If the DMA spans multiple states, define clusters as DMA-states as long as the smaller sub-market contains at least 4 stores.
2. If that fails, define clusters as DMA-counties, as long as the smaller sub-market contains at least 4 stores.
3. If that fails, define clusters as DMA-zip3 as long as the smaller sub-market contains at least 4 stores
4. If that fails, split stores randomly into two sub-markets.

In variations 1-3, store clusters are aggregated into sub-markets works as follows:

1. Assign the largest cluster to sub-market 1
2. Assign the second-largest cluster to sub-market 2
3. Work through the remaining clusters in descending size (third-largest cluster first) and assign each cluster to the sub-market that currently has fewer stores.

A.1.11 Empirical Bayes Procedure

To adjust for sampling error in our estimates $\hat{\eta}_{sj}$ of individual store-product level elasticities, we apply an Empirical Bayes (EB) correction following the approach that has become standard in the education literature (Kane and Staiger, 2008; Jacob and Lefgren, 2008; Angrist et al., 2017). We define EB-adjusted elasticities to be

$$\tilde{\eta}_{sj} = \left(\frac{\sigma_{rj}^2}{\sigma_{rj}^2 + Var\left(e_{sj}\right)} \right) \hat{\eta}_{sj} + \left(\frac{Var\left(e_{sj}\right)}{\sigma_{rj}^2 + Var\left(e_{sj}\right)} \right) \overline{\eta}_{rj},$$

where $\overline{\eta}_{rj}$ and σ_{rj}^2 are a prior mean and variance defined at the chain-product level and $e_{sj} = \hat{\eta}_{sj} - \eta_{sj}$ is the estimation error in $\hat{\eta}_{sj}$.

We define $Var\left(e_{sj}\right)$ to be the estimate of the asymptotic variance of $\hat{\eta}_{sj}$ from the regression in equation (3). Recall that these regressions are run at the store-product level and the asymptotic variance is clustered by two-month periods. We define the hyperparameter $\overline{\eta}_{rj}$ to be the simple average of $\hat{\eta}_{sj}$ within chain r and product j. We define the hyperparameter σ_{rj}^2 to be an estimate of the variance of η_{sj} across stores s within chain r and product j, which we form by computing the variance of $\hat{\eta}_{sj}$ across stores within chain r and product j and then subtracting the mean of $Var(e_{sj})$ across stores within chain r and product j. In a small number of cases where this yields a negative estimate of σ^2_{rj} we set $\sigma^2_{rj} = 0$.

The adjusted elasticities are used for (i) plotting the distribution of elasticities in Figure VII Panel B; (ii) computing lost profits in Table IX; (iii) estimating average marginal cost for the analyses in figure XI and Table XI. All other analyses in the main paper use raw elasticities.

A.1.12 Event Study

In this Appendix, we provide the exact criteria by which we identify the sets of stores that switch owner, as well as how we identify the timing of the switch.

We identify switching stores as ones that switch once from one *parent_code* to another *parent_code* within the 2006-2014 sample. We consider as switchers only stores that change parent_code and not ones that change *retailer_codes* within a *parent_code*. Then, we identify switching cohorts or episodes as incidents where at least two stores switch from one *parent_code* to another *parent_code*.

To identify the timing of when a set of stores switches *parent_code*, we define a measure of assortment similarity between these switching stores and their respective old and new *parent_codes*. We compute this measure for two modules, the orange juice module and the cereal module. Specifically, for each quarter the measure of assortment similarity to the old (new) *parent_code* is the share of products sold for at least 5 weeks by the old (new) *parent_code* that are also sold by the switching stores for 5 weeks in that quarter. Then, we define switch t_0: if the switching stores close temporarily during the transition, then we take the midpoint of the quarters closed; if the switching stores do not close, then we take the first quarter when the assortment similarity to the new *parent_code* is greater the assortment similarity to the old *parent_code*; if the assortment similarity crosses more than once, then we drop these switchers. We compute the switch timing t_0 for the two modules, and we keep only the switchers such that the two modules identify the same t_0.

We define the pre-period as the quarters up to one quarter before t_0. The post-period starts the first quarter after t_0 where the assortment similarity to the new *parent_code* has closed 75% of the average gap between the assortment similarity to the old and new *parent_codes* during the pre-period. If this convergence takes longer than three quarters, we do not consider these switching stores to have a valid post-period. This definition of the post period does not depend on the assortment similarity to the old *parent_code* in the post-period. Mergers where the old chain closes all stores is still considered a valid merger as long as the assortment similarity to the new *parent_code* converges sufficiently quickly.

For the analysis of pricing similarity between stores, we define a measure similar to the measure of quarterly absolute price difference for store pairs, except that (i) it is computed using weekly, as opposed to quarterly, prices, (ii) we standardize the week t such that the week of the switch is week 0, and (iii) instead of comparing stores s and *s'* in a pair of stores, we instead compare each store-product-week *sjt* for the switching stores to the average log price for product j in week t in non-switching stores of the "old" (respectively, "new") chain. We aggregate to the weekly level taking the simple average of this measure across products j, and then the simple average across the switching stores s.

In order to compute longer-run elasticities, we keep up to 52 weeks of data from the start of the post-period and up to 52 weeks prior to the end of the pre-period. At minimum, there are three quarters of data available in the post-period for each episode. We sample up to 200 stores from the old and new *parent_code* as control stores. For the pre-specification and post-specification, we estimate equation (6) keeping only the old *parent_code* stores and new *parent_code* stores as controls, respectively.

A.1.13 Inequality

In our income inequality exercise, we suppose that there is a representative product sold by every store with a marginal cost, c, constant across chains. Per our model, the optimal flexible pricing is $p_s^* = \lambda_s + log(c)$, where $\lambda_s = log(\frac{\eta_s}{1+\eta_s})$. We make two further assumptions: 1) the value of c is set equal to the median of the empirical estimates of marginal cost, $\hat{c}_{rj}$; 2) the value of λ_s is set according to the income first stage, exactly as in Table V, column 5. We add the overall mean of $\hat{\lambda}_{sj}$ for level.

Define the uniform log price $p_r^{Uniform}$ as the log of the chain-level uniform price that optimizes the profit equation $\sum_s k_s P_r^{\eta_s}(P_r - c)$. We assume that stores are all the same size with $k_s = 1$ for all s.

The yearly log price paid perturbs the uniform log price within each chain by the yearly IV coefficient, β^{Yearly}, of prices on elasticity (Table VIII, column 1) which is meant to include the "automatic stabilizer" effect of intertemporal substitution due to sales. Formally, the yearly log price paid is $p_s^{Yearly} = p_r^{Uniform} + \beta^{Yearly}(\lambda_s - \bar{\lambda}_r)$, where $\bar{\lambda}_r$ is the average of λ_s within chain r.