For Online Publication

The Elusive Costs of Inflation:

Price Dispersion during the U.S. Great Inflation

Emi Nakamura
Columbia University

Jón Steinsson
Columbia University

Patrick Sun
FCC

Daniel Villar
Federal Reserve Board

May 26, 2018

A Derivations

A.1 Flexible Price Case

The period t profits of intermediate firm i are given by

$$\Pi_{it} = p_{it} y_{it} - W_t L_{it}. \tag{1}$$

Using the demand curve the firm faces—equation (4) in the main text—and the firm's production function—equation (8) in the main text—we can rewrite the profit function as

$$\Pi_{it} = p_{it} \left(\frac{p_{it}}{P_t} \right)^{-\theta} C_t - \frac{W_t}{A_{it}} \left(\frac{p_{it}}{P_t} \right)^{-\theta} C_t. \tag{2}$$

Maximization of this expression as a function of p_{it} yields

$$p_{it} = \frac{\theta}{\theta - 1} \frac{W_t}{A_{it}}. \tag{3}$$

Raising the expressions on both sides of this equation to the power $1 - \theta$ and integrating over i yields

$$\int_0^1 p_{it}^{1-\theta} di = \left(\frac{\theta}{\theta - 1} W_t \right)^{1-\theta} \int_0^1 A_{it}^{\theta - 1} di. \tag{4}$$

Raising both sides of this expression to the power $1/(1-\theta)$ yields

$$P_t = \frac{\theta}{\theta - 1}\frac{W_t}{A_f}, \tag{5}$$

where A_f is given by

$$A_f = \left[\int_0^1 A_{it}^{\theta-1} di\right]^{\frac{1}{\theta-1}}. \tag{6}$$

Combining firm i's production function—equation (8) in the main text—and the demand curve for firm i's output—equation (4) in the main text—yields

$$L_{it} = \left(\frac{p_{it}}{P_t}\right)^{-\theta} A_{it}^{-1} C_t. \tag{7}$$

Integrating over i and using firm i's price setting equation—equation (3)—yields

$$\int_0^1 L_{it}\, di = \left(\frac{\theta}{\theta - 1}\frac{W_t}{P_t}\right)^{-\theta} \int_0^1 A_{it}^{\theta-1} di\, C_t. \tag{8}$$

Using equation (5), this equation can be simplified to yield

$$Y_t = A_f L_t, \tag{9}$$

where Y_t denotes aggregate output and we have $Y_t = C_t$.

A.2 Sticky Price Case

As in the flexible price case, combining firm i's production function—equation (8) in the main text—and the demand curve for firm i's output—equation (4) in the main text—yields

$$L_{it} = \left(\frac{p_{it}}{P_t}\right)^{-\theta} A_{it}^{-1} C_t. \tag{10}$$

In this case, however, some labor is potentially used to change prices. Let's denote this by L_t^{pc}. Taking this into account and integrating the above expression over i, we get that aggregate labor supply is

$$L_t = \int_0^1 \left(\frac{p_{it}}{P_t}\right)^{-\theta} A_{it}^{-1} di C_t + L_t^{pc} \tag{11}$$

Rearranging this equation and using the fact that $Y_t = C_t$, we get that

$$Y_t = A_t(\bar{\pi})(L_t - L_t^{pc}), \tag{12}$$

where

$$A_t(\bar{\pi}) = \left[\int_0^1 \left(\frac{p_{it}}{P_t}\right)^{-\theta} A_{it}^{-1} di\right]^{-1}. \tag{13}$$

B Data Appendix

The process of scanning the microfilm cartridges left us with about 600 image folders—one corresponding to each cartridge. Each folder contains roughly 2000 images for a total of about 1 million images. The images are called Price Trend Listings and contain a table of data regarding several products (the rows in the table) over a 12 month period (the columns in the table). All data on each image comes from products in a particular product category (ELI). The left most column on each image contains a location (city) identifier, an outlet identifier, a product identifier, and a version identifier. The top row lists the time periods to which the data refers. The right most column contains the data for the month the images was created in. The remaining 11 columns repeat older data on the same products—i.e., show the "price trend" for each product. We refer to the month the image was created in as the "collection period" for this image. Each of the interior cells in the table is divided into three sub-cells. The top sub-cell reports the price. The middle sub-cell reports a number of flags including a sales flag and a flag related to whether the price is imputed. The bottom sub-cell contains the percentage change of the price since the last collection period for that product. Within each collection period, images are sorted by product category (ELI). Within each image, rows are sorted by the values of the identifiers in the left most column. They are first sorted by the outlet identifier, then by the product identifier, and finally by the version identifier.

As we describe in the main text, we used optical character recognition (OCR) software to convert the scanned images to machine readable form. We worked closely with a software company to create custom software that could convert the Price Trend Listings as accurately as possible. While we were able to find ways to eliminate most common systematic errors that we came across (especially in the price variable), it is inevitable given the current state of OCR technology that there are some random errors that remain in the raw data that results from the OCR process. Fortunately, the format of the Price Trend Listings described above implies that there is a large amount of redundancy in the raw dataset. This redundancy can be used to validate the output of the OCR process. Below we describe the procedure we use to validate the output from the OCR procedure.

B.1 Product Categories and Product Identifiers

The first step is to validate and improve on the OCR output for category labels and product identifiers. We first set to missing all ELI values that do not correspond to one of the values on the list of ELI values in the BLS classification. We then use the fact that the images are ordered by ELI within each collection period to fill in missing ELI information in cases where the last observed ELI value and the next observed ELI value are the same. Finally, in cases where there are large blocks of images that still have missing values for the ELI, we manually review the original scanned images to determine which image separates the different ELIs. By these steps we are able to validate the ELI for 99.7% of the images we have.

We use a similar procedure if there is a missing value of a product-identifier. We use the fact that the images are sorted by ELI and the observations within image are sorted first by location, then by outlet, then by product, and finally by version. First, we set identifiers that are out of order to missing. We then fill in identifiers in cases where the identifier before and after a block of observations with missing values for the identifier are the same.

Errors in reading product-identifiers lead to spurious products in our dataset. Whenever such errors occur, an entire row in a single Price Trend Listing image will be associated with this "phantom product" (i.e., this erroneous product code). We will therefore have up to 12 months of price data for these phantom products, which will result in up to 11 months of price change statistics. If the distribution of phantom products is non-uniform across product categories, their presence could bias our results by putting more weight on product categories where there are more phantom products. However, it is unlikely that exactly the same error will occur for multiple images with the same product-month. This implies that product months of phantom products will appear only on a single image (i.e., we won't have more than one replicate for product-months of phantom products). This, in turn, implies that our first algorithm for accepting price observations in to the final dataset (described below) will not accept these observations. We have rerun our results with only data accepted by the first algorithm and they are virtually identical. This makes us confident that the phantom products are not biasing our results.

The presence of the phantom products does artificially inflate the number of observations in the dataset. Whenever the same observation appearing on more than one image is read in different ways from different images, what is suppose to be a single observation, turns into more than one

observation. We see signs of this occurring in the early part of our sample and in particular in the latter half of 1979 and the first half of 1980. Over this period, the number of observations per month rises to 170,000 per month. But the number of observations that appear on only a single image also rises very substantially (to over 50,000 per month). We take this as a sign that in this period a substantial number of phantom observations are appearing in the dataset.

B.2 Prices

We use two main procedures to validate the OCR output for prices. First, since each Price Trend Listing contains not only the price for that period but also prices for up to 11 earlier month, each product-month observation may appear multiple times in the raw dataset. For example, suppose we consider a product that the BLS collected a price for from October 1979 to November 1980. The October 1979 price will appeal on an image in October 1979 and will then be repeated on images in each of the subsequent 11 months. The October 1979 price of this product will therefore show up 12 times in the dataset. We refer to these 12 instances as 12 replicates of the same product-month observation.

Most product-month observations will have fewer than 12 replicates. Our data is based on monthly micro-film cartridges from May 1977 to December 1980 and bimonthly micro-film cartridges from January 1981 onward. Product-month observations in the later part of our dataset will therefore appear at most 6 times. Also, certain products in certain cities are sampled only bimonthly. These will also only appear at most 6 times. Finally, product-month observations towards the end of a product's life-time in the BLS sample may appear less often. But even the last observation for a product in many cases appears more than once. Figure 1 plots the distribution of number of replicates in our sample.

We use this redundancy to verify the accuracy of the OCR output for prices. Our rule is to accept the price that we observe most often among the replicates for each product-month as long as we observe this price at least twice. If no price is observed more than once, we don't accept any price using this algorithm (and instead rely on the 2nd algorithm described below). It is very rare that more than one price is observed at least twice. This occurs for only 0.04% of product-months. It is even rarer, that there are more than one prices that are observed at least twice and an equal number of time, i.e., that there is a tie as to which price is observed most often. This occurs for only 0.004% of product-months. In these cases, we accept neither price (and again rely on the 2nd

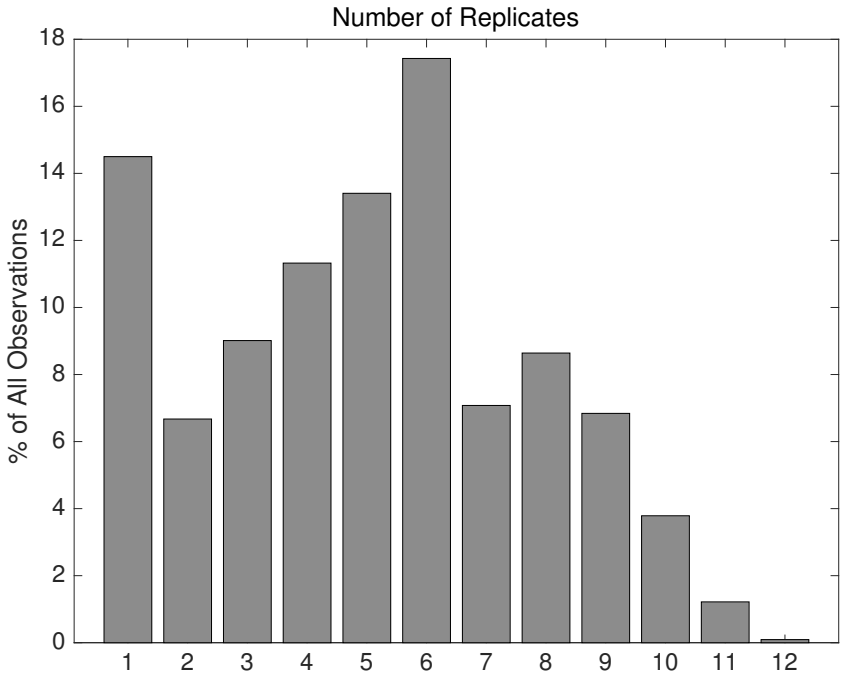

Figure 1: Distribution of Number of Replicates for Each Product-Month Observation

algorithm). Overall, prices are accepted using this algorithm for 86.5% of product-months.

Once we have finish running this first validation algorithm, we take all replicates for product-months for which prices have been accepted and we fill in the accepted price. In other words, we "correct any errors" in all the replicates for the product-months for which the first algorithm accepts observations. We do this in order to improve the chances that the second algorithm is able to validate prices for additional product-months.

The second redundancy we make use of is the percentage change variable. We calculate the percentage change in the price for a particular product-month from the price we observe on the image in that month and the price we observe on the image in the previous month, whenever both are present on the same image. We round this calculated percentage change to the next whole percent. If the value we calculate for the percentage change in this way matches the value reported in the percentage change variable on the image, we accept the price for both the product-month in question and the previous product-month (i.e., both price values used to calculate the percentage change).

The second way in which we use the percentage change variable is that when the following conditions are met:

- The percentage change in the price that we calculate from the price observations does not line up with the percentage change variable read directly from the image in a particular product-month and also does not line up for the same product in the next month.

- The percentage change read from the image for the product-month in question and both the next and last month for that product are all equal to zero

we set the price in the product-month to the value read in the previous month and accept this value into our main dataset. This is meant to catch cases where the OCR software made an error in the price variable, but the percentage change variable gives us a strong indication that the price actually stayed constant.

The procedure we describe above that uses the price change variable on the Price Trend Listing images is repeated for each image that a particular product-month observation occurs on. A price for the same product-month can therefore be accepted more than once (in principle as often as a particular product-month occurs on different images). It is even possible that two or more different prices are accepted for the same product month using this procedure. This only occurs for 0.14% of observations. Whenever this occurs, we drop all accepted observations based on the percentage change procedure. Overall, these procedures for using the percentage change variable raises the overall acceptance rate to 98.4% of the observations in our raw dataset. We drop the remaining observations.

B.3 Price Flags

As we discuss above, the second sub-cell of each cell in the Price Trend Listings images contains a string of characters which code various information regarding the price in question. This string contains at most seven characters. In some cases, some of these characters are left blank. The first three spots are reserved for characters indicating, among other things whether a product is on sale, whether it was unavailable, whether it is a seasonal item, etc. Typically, at most one of these spots will contain a letter and the others will be left blank. The next three spots give information about which pricing cycle the product belongs to (some products are priced monthly and others bimonthly). The last spot may contain the letter "I" indicating that the price was imputed. If the price was not imputed, this spot is left blank. We had the OCR software convert blank spots to # signs to help us tell which spot of the string each character occurred in. However, this conversion

was somewhat imperfect.

Our OCR procedure turned out to be less accurate in converting these price flags, but fortunately the price flags are chosen from a restricted set of characters and the errors follow well-defined patterns. Our interest centers on identifying two pieces of information from this string of characters: 1) whether the product was on sale, 2) whether the price was imputed.

The letter "B" is the character that indicates that the product was on sale. We therefore create a sales flag variable and set it to 1 if the letter "B" occurs in the string of characters. Due to worries about accuracy or the OCR procedure for these flags, we manually compared the OCR output with the corresponding images for a subset of the images. This process revealed that the OCR process sometimes converted the letter "B" in the string to "6", "8", "9", "0", "O". We therefore set the sales flag to 1 whenever we observed any of these characters in the string.

As we mention above, the letter "I" in the last spot of the string indicates that the price was imputed. We therefore create an imputation flag variable and set it to 1 if the letter "I" is observed at the end of the string. Our manual comparison of the OCR output and the corresponding images revealed that in some cases the letter "I" was read as "1" by the OCR procedure. The number "1" also appears as a part of the pricing sample part of the string. But our manual inspection indicated that the number "1" appearing at the end of the string and being preceded by another number, gave strong indication that the price was imputed. In these cases, therefore, we set the imputation flag to one.

In addition to this, several of the characters appearing in the first three spots of the string signal that the price was imputed. These include "A" (seasonal item not available), "C" (closeout or clearance sale)[1], "D" (ELI not available), "U" (unable to price), "T" (temporarily unable to price). The first three of these (A, C, D) may also appear in the pricing cycle portion of the string. We therefore set the imputations flag to one whenever we observe these characters in one of the first three spots of the strings. The other three characters should not appear in other parts of the string. We therefore set the imputation flag to one whenever we observe these characters.

As in the case of prices, a price flag should appear in multiple "replicates" due to the structure of the Price Trend Listing, and in principle, these replicates should be the same. In cases where they disagree, we choose the value of the flag corresponding to the majority of replicates. This

[1]BLS documentation we received indicated that "C" referred to "closeout or clearance sales" but out inspection of a subset of images indicated that these observations were imputed (e.g., they tended to include more than two numbers after the decimal place (fractions of a cent).

rule is meant to balance a concern for false positives and false negatives based on our inspection of a subset of cases where is disagreement. The fraction of observations that we drop because of imputation is 9.3%.