Seasonality, Consumer Heterogeneity and Price Indexes: The Case of Prepackaged Software *

Adam Copeland†

January 5, 2011

Abstract

This paper measures constant-quality price change for prepackaged software in the US using detailed and comprehensive scanner data. Because there is a large sales surge over the winter-holiday, it is important to account for seasonal variation. Using a novel approach to constructing a seasonally-adjusted cost-of-living price index that explicitly accounts for consumer heterogeneity, I find that from 1997 to 2003 constant-quality software prices declined at an average 15.9 percent at an annual rate. As a point of comparison, the Bureau of Labor Statistics reports average annual price declines of only 7.7 percent for prepackaged software.

Keywords: seasonal adjustment, software prices, heterogeneity, price indexes

*I would like to thank Dennis Fixler and Marshall Reinsdorf for helpful comments. I also thank Andrew Miller for substantive help as a research assistant. The views expressed herein are my own and not necessarily those of the Federal Reserve Bank of New York or the Federal Reserve System.

†Money and Payments Studies, Federal Reserve Bank of New York, 33 Liberty St., New York, NY 10045; e-mail: adam.copeland@gmail.com; webpage: http://www.copeland.marginalq.com

1 Introduction

Software plays an important role in the information technology revolution that has swept the US. Yet compared to other information technology products such as computers and semiconductors, relatively little research has been devoted to understanding this sector.[1] This paper aims to help fill this gap by constructing a price index for prepackaged software typically sold to consumers.[2] A better understanding of software pricing trends is an important component of the measurement of consumer durable goods and ultimately of consumer welfare.

Using scanner data from the NPD Group to construct a maximum-overlap Fisher price index, I find the average price decline for prepackaged software is 14.7 percent at an annual rate. In the data, however, I find the majority of software products experience a significant boost in unit sales in December, the main month in the US winter-holiday season. Given there is seasonality in the data, Alterman et al (1999) claim the Mudgett-Stone index (i.e. a year-over-year approach) should be considered the best measure of annual price change (page 48). The underlying framework of the Mudgett-Stone approach, however, is a representative-consumer framework.[3] In the prepackaged software market, the winter-holiday variation seems to be driven by consumer heterogeneity. Given the particular correlations of prices and sales over these months, I argue that the arrival of casual, once-a-year shoppers in December are the main driving force behind the surge in winter-holiday sales. Hence, there are two main types of consumers in this market: regular shoppers who purchase prepackaged software year-round, and casual, once-a-year shoppers who only purchase prepackaged software in December (as part of the winter-holiday season).

Using this insight, a better way to account for seasonality in the prepackaged software market is to explicitly account for this consumer heterogeneity by constructing two indexes: one index for the casual, once-a-year consumers, and another index for the regular, year-round shoppers. These two indexes are then averaged using revenue weights,

[1] Jorgenson (2001) emphasizes that information gaps remain about understanding software pricing trends.

[2] See Parker and Grimm (2000) for details on the high rate of growth of prepackaged software.

[3] See Diewert (1998, 1999) and Nesmith (2007).

resulting in what I label the Heterogeneous index. For the software market as a whole, this price index measures the average constant-quality price decline to be 15.9 percent at an annual rate. Using the standard Mudgett-Stone approach, constant-quality annual price change averages 17.4 percent. Hence, properly accounting for the underlying consumer heterogeneity lowers the measured fall in prices over the sample by 1.5 percent at an annual rate, not an inconsequential difference. Naturally, there are larger differences at lower levels of aggregation. For the software categories PC Games, Finance and System Utilities, the differences in estimates of constant-quality annual price change between the Heterogeneous index and the Mudgett-Stone index are 3.5, 7.2, and 11.0 percent, respectively.

From these results, I draw two main conclusions. First, whether or not you account for seasonal variation, prepackaged software prices are declining at an annual rate of at least 15 percent between 1997 and 2004. This is more than two times as large as the 7 percent average annual rate of decline reported by the Bureau of Labor Statistics (BLS) for this same period. As detailed later, the difference in measured price change between the BLS and this paper's price indexes is driven, in roughly equal parts, by differences in data and the method of index construction.

Second, while there is little doubt that accounting for seasonal variation is important, the results above demonstrate the importance of understanding the underlying causes in the variation. If seasonal variation is not driven by consumer heterogeneity, then the Mudgett-Stone price index is likely the best available approach. But if consumer heterogeneity is driving the change in price and quantity sold over the year, then the Mudgett-Stone price index can be misleading. I argue that the Heterogeneous index presented here is a more appropriate measure of price change.

The results from the Heterogeneous index, however, should be taken with a note of caution. A crucial step in constructing this index is determining units sales by each type of consumer. Ideally, there would be sales data by type of consumers. Since the consumer types are not observed, however, I use an *ad hoc* approach to divide unit sales over the winter-holiday between the two consumer types: the casual, once-a-year shoppers and the regular, year-round shoppers. Reassuringly, the results are robust to alternative approaches to dividing up unit sales.

A number of researchers have already produced price indexes for software; this paper builds upon this small literature in two main ways. First, unlike most previous work, this paper uses detailed, industry-wide scanner data, as opposed to a small subset of products.[4] Hence, the price indexes are representative of price changes throughout the industry, and so generate robust measures of constant-quality price change. Second, I develop and implement an approach to constructing price indexes that accounts for the large amount of seasonality within the software industry. Because previous software price indexes have ignored seasonality, the inclusion of seasonality adjustment is, in itself, a modest improvement on previous empirical work. I also introduce a new variation on the existing set of empirical methods used to account for seasonality when constructing a price index. Unlike previous methods, this approach explicitly accounts for consumer heterogeneity, the driving force behind prepackaged software's seasonal fluctuations.

This paper is closest to Prud'Homme et al (2005), who construct a maximum-overlap Fisher price index for prepackaged software using samples of transaction level data from the Canadian market. This paper's results on the average annual decline in constant-quality price differ substantial from Prud'Homme et al (2005), however. They report an average annual price decline of 7.9 percent. Using their price index methodology on the NPD scanner data, I find an average annual price decline of 14.7 percent. Hence differences in the data must be driving the result. The NPD scanner data, while not the universe, is more comprehensive than their sample. Further, the data are detailed enough information that different versions of the same software product are observed. Besides the difference in empirical results, the current paper differs from Prud'Homme et al (2005) because it accounts for the seasonality in the data.

By accounting for consumer heterogeneity, this paper builds upon the work of Griliches and Cockburn (1994), Fisher and Griliches (1995), and more recently, Aizcorbe et al (2010) and Aizcorbe and Copeland (2007). These works consider the effects of consumer heterogeneity on the construction of price indexes with respect to the introduction of new goods. This paper differs from these works because of its focus on seasonality.

[4]For example, Oliner and Sichel (1994) and McCahill (1997) study price movements of word processors, spreadsheet, and database software applications, Abel et al (2003) examine price movements of Microsoft's personal computer software products, and Gandal (1994) analyzes prices of spreadsheets.

2 Data

In this section I describe the data on prepackaged software. I then measure the seasonality within the data and discuss two data quality issues.

2.1 Description

The prepackaged software industry data come from the NPD Group.[5] Software is prepackaged when it is sold or licensed in standardized form and is delivered in packages or as electronic files downloaded from the Internet. This is opposed to custom and own-account software which require larger degrees of tailoring to the specific application of the user.[6] The data are point-of-sale transaction data (i.e. scanner data) that are sent to the NPD Group from participating outlets. The data purchased from the NPD Group are retail sales, or transactions from warehouse club stores, internet retailers, office superstores, etc. NPD claims to cover 84 percent of the U.S. retail market, and so provides a clear picture of the prepackaged software retail market. The data are monthly observations at the national level, where a record is a product. A sample product is "Barney Goes To The Circus," a software program published by Microsoft. For each observation, the revenue earned and the number of units sold that month are reported, allowing me to compute the average monthly price of the product. Further, the data include the name of the software publisher, and category and subcategory variables that provide an classification structure for grouping products. The time frame of the data ranges from January 1997 to August 2004 and includes 782,849 observations. Table 1 provides a summary of the data at the category level, showing the number of subcategories and observations within each category as well as the relative size of each category by units sold and revenue generated. PC Games is the largest category by far, accounting for 35 percent of total revenue and almost half of all sales. Business and Finance, are the next two largest categories and together account for roughly 25 percent of total revenue generated within this market.

[5]For information on this marketing and consumer research firm go to http://www.npd.com .

[6]These definitions follow those used to measure prepackaged software in the U.S. national income and product accounts. See Parker and Grimm (2000) for more details.

Category	Subcategories	Observations	Unit sales (millions)	Revenue (millions)	Suppressed (percent)
Business	23	108,216	133	12,940	0.0037
Education	30	150,213	449	9,633	0.0009
Finance	3	13,239	262	11,985	0.0002
Imaging/Graphing	16	76,341	195	8,861	0.0009
Operating System	3	14,068	71	7,032	0.0004
PC Games	13	294,243	1,482	34,505	0.0002
Personal Productivity	33	75,637	183	5,910	0.0017
System Utilities	25	50,892	207	9,754	0.0011
Total	146	782,849	2,983	100,619	0.0007

Table 1: Data Summary

The mean lifespan for an average software product is 22.0 months. This statistic is skewed by a few extremely longed-lived products; the median length of time an average product is sold in the market is 17 months. As shown in Table 2, there is a large amount of variation in the length of time a product is sold by category. The median number of months a product is sold ranges from 9 to 35 months, where System Utilities products have the shortest average lifespan and PC Games the longest. The 22 month lifespan of the average prepackaged software product, however, is slightly deceiving. On average, a software product generates 75 percent of its lifetime revenue in the first year of its life. Hence, the tail end of software product's lifespan tends to be unimportant.

Behind this last fact is the declining trend in both price and units sold for prepackaged software over the product cycle. To measure how quickly price and units sales fall over the product cycle, we regressed the log of these variables on product cycle dummy variables, with fixed effects for each software product and using revenue weights. The estimated coefficients for the product cycle dummy variables, which are all precisely estimated, are graphed in Figure 1.[7] The results indicate that prices fall over 19 percent over the first year a product is sold while unit sales decrease 50 percent. Hence, prepackaged software is a market where, over the product cycle, prices are rapidly falling alongside plummeting unit sales.

[7] In the appendix, we report the coefficient estimates and associated standard errors.

Category	Mean	Median
Business	15.5	10
Education	27.8	26
Finance	23.0	19
Imaging/Graphics	19.3	14
Operating System	16.1	13
PC Games	34.5	35
Personal Productivity	27.7	26
System Utilities	13.7	9
All	22.0	17

Table 2: Prepackaged Software Life (Months)

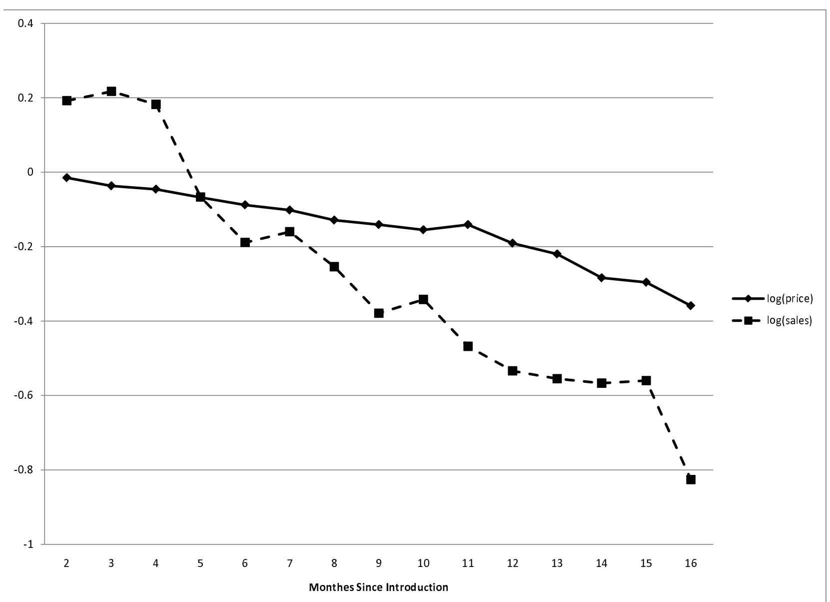

Figure 1: Price and Sales Contours over the Product Cycle

Notes: The figure plots coefficients estimated from regressing the log of price and the log of sales on product cycle dummies, with product-level fixed effects.

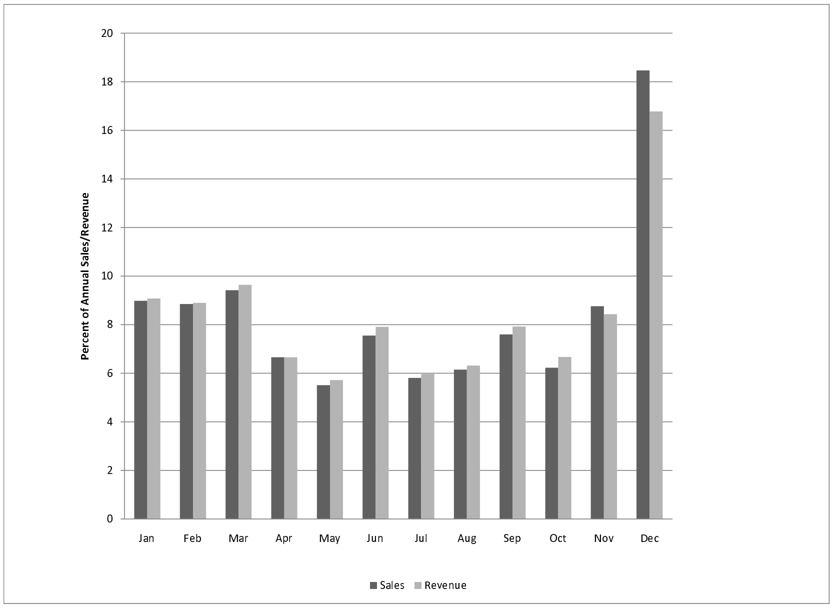

Figure 2: Percent of Units Sold and Revenue Generated by Month

Note: Results computed using data from Jan 1997 to Dec 2003.

2.2 Seasonality

A priori it is not surprising that some products within the prepackaged software market exhibit strong seasonality over the winter holiday. The rise in U.S. retail sales over the winter holiday is well-known phenomenon.[8] Looking at the raw prepackaged software monthly data, it is not hard to see a significant winter holiday sales surge across many prepackaged software categories. Figure 2 charts the percentage of units sold and revenue generated by month for all prepackaged software from January 1997 to December 2003. Clearly, December is a significant month for publishers of software, contributing over 18 percent of units sold annually and almost 17 percent of total revenue for the year.

[8]The U.S. Census Bureau publishes retail sales seasonal factors which show the large surge in sales in December for most kinds of businesses (see http://www.census.gov/svsd/www/adseries.html).

Identifying which products experience a winter-holiday seasonal effect is complicated by prepackaged software's short-lived product cycle. As described above, the median length of time a specific software product is sold is 17 months. Further, the vast majority of the revenue that software generates occurs within the first year, devaluing year-over-year comparisons. Hence, for the majority of cases, I am not able to definitively determine if there is a winter holiday seasonal effect at the product level.

To identify seasonal effects I consider the data at a higher level of aggregation–the subcategory level.[9] Several approaches were used to determine when a subcategory of software experiences a winter-holiday seasonal affect. The preferred approach, and the one presented here, uses x-12-ARIMA, a seasonal adjustment software packaged used and maintained by the U.S. Census Bureau.[10]

The x-12-ARIMA algorithm produces a seasonally-adjusted series of units sold for each subcategory of software. For each subcategory and for each year, I state there is a winter-holiday seasonal effect when December units sales in the seasonally-adjusted units sold series are less than December unit sales in the non-seasonally-adjusted series.

Using this framework, I find that winter-holiday seasonality is pervasive in the prepackaged software market. Across all subcategories, the median value of the ratio of seasonally-adjusted to non-seasonally-adjusted units sold is 0.59 in December. For all other months except March, the median values of this ratio greater than or equal to 1. The seasonality in March is partly driven by income-tax preparation software.

I denote the difference between the non-seasonally-adjusted and seasonally-adjusted revenue series as the "seasonal" component of revenue. The magnitude of this winter-holiday seasonality is substantial, accounting for 48 percent of total December revenue. Seasonal revenue differs substantially across types of software, ranging from 15 to 61 percent of total revenue (see Table 3). Business software has the smallest seasonal component, in-line with priors that work-related software is not much affected by the winter-holiday season. Education and PC Games software have the largest winter-holiday effect; for these categories, more than half of total revenue in December is attributable to the seasonal component.

Using x-12-ARIMA, or any statistical approach, to define the seasonal component

[9] An example of a subcategory is "Foreign Language" software within the "Education" category.

[10] See http://www.census.gov/srd/www/x12a/ for more information.

Category	Seasonal Component (percent)
Business	15.2
Education	55.8
Finance	25.9
Imaging/Graphics	39.3
Operating System	28.6
PC Games	60.9
Personal Productivity	45.4
System Utilities	28.1
All	48.0

Table 3: Seasonal Revenue as a Percent of Total December Revenue

of revenue at the sub-category level is complicated by the endogenous entry problem. Software publishers regularly release both completely new software programs as well as updated versions of current software. In an average month, 4 percent of the software products sold have just entered and 4 percent are exiting.[11] Given the general rise in demand over the winter-holiday, firms may have an incentive to introduce new products over the winter-holiday to take advantage high demand. Indeed, examining the lifetime sales weighted average of product introductions by month, I find that a disproportionate amount of products are introduced in September, October and November, around the beginning of the fourth quarter (see Figure 3). Product exits also follow a seasonal pattern, with a disproportionate number of exits occurring in December (see Figure 4). Because unit sales and revenues at the end of a software's product cycle are small, the strongly seasonal nature of exits is surprising. Perhaps the high demand in December allows retailers to sell off the remaining inventory of older products.[12]

In part because I aggregated the data to the subcategory level, the x-12-ARIMA approach does not take entry into account when it computes seasonal factors. As such we cannot distinguish how much of the winter-holiday sales surge is due to the introduction of new products and how much is a purely seasonal effect. Properly dealing

[11]I define entry as the first month a product appears in the data and exit as the last month a product appears in the data.

[12]The large amount of entry and exit in March is partly driven by income-tax preparation software.

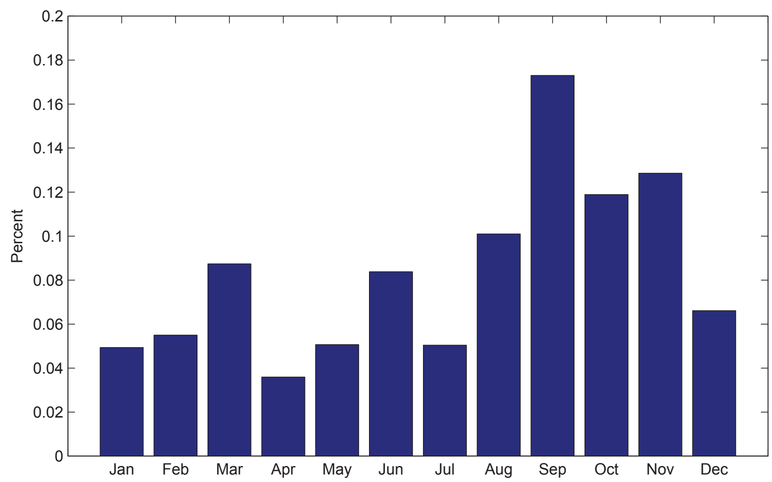

Figure 3: Product Introductions by Month

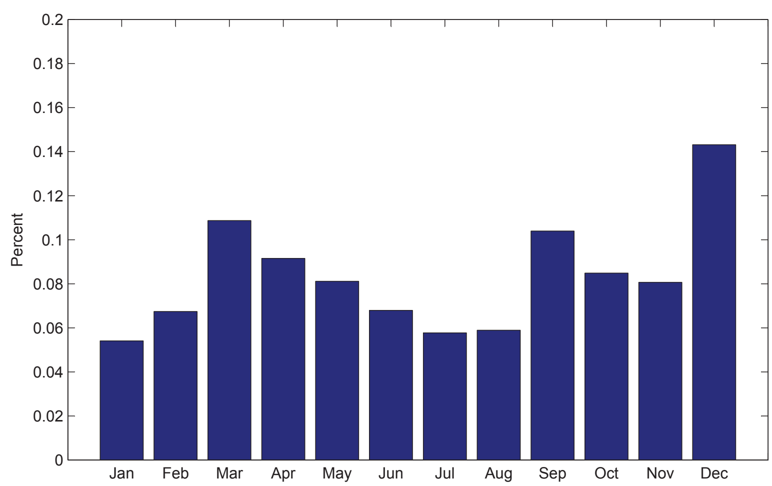

Figure 4: Product Exits by Month

Quantile	99%	95%	90%	75%	50%	25%	10%	5%	1%
Price Ratio	5.00	1.66	1.24	1.03	1.00	0.92	0.70	0.50	0.18

Table 4: Frequency Distribution of the Price Ratio of Adjoining months' Prices

with this endogenous entry problem requires a formal model of both consumer demand and publisher profit-maximization, something beyond the scope of this paper. Instead, consider the seasonal factors produced by the x-12-ARIMA program to be good, first-cut approximations.

2.3 Data Quality

Before discussing how to construct price indexes that adjust for the winter-holiday seasonality, two quality issues in the data are addressed. First, observations are suppressed by the NPD Group whenever a product's sales for a particular month come from fewer than five retailers. NPD aggregates these suppressed data together into a single observation by subcategory. Because this aggregation mixes products inconsistently over time, these observations are excluded from the analysis. As shown in the last column of Table 1, these observations account for a negligible share of the total units sold.[13]

Second, there are implausible monthly price changes. As shown in Table 4, the price ratio of adjoining months' prices has some extreme values. Categorizing which monthly price changes are the result of measurement error can be difficult to discern. I take a conservative approach and drop the observations that are in the top and bottom five percent of the monthly price ratio distribution. This translates into dropping monthly price ratios below 0.50 and above 1.66.[14]

[13]In addition to the removing the suppressed observations, I also removed four subcategories in which the percentage of suppressed observations accounted for over 60 percent of units sold. These subcategories are Data Center Management, Drivers/Spoolers, Engineering, and Network Resource Sharing, and together they make up an insignificant portion of all units sold.

[14]The main results of the paper are robust to only dropping monthly price ratios that are in the top and bottom one percent. In this case, however, some of the price indexes for Finance software are implausibly high.

3 Methods

In this section I describe and compare three different price index approaches: the maximum-overlap Fisher, the Mudgett-Stone and the Heterogeneous price indexes.

3.1 Fisher price index

The maximum overlap Fisher price index is a standard approach to measuring constant quality price change. This is the approach used by Prud'Homme et al (2005) in their work measuring price change for prepackaged software. The Fisher is an average of a Laspeyres and Paasche price index. I use a moving basket of goods and so the reference period when considering month t is month $t-1$. Denote $\mathcal{L}_t^{c,\text{std}}$ as the Laspeyres monthly price relatives for software in group c and $\mathcal{P}_t^{c,\text{std}}$ the Paasche. Letting $J_{t,s}^c$ be the set of products belonging to the software group c available in both month t and s, I compute the Laspeyres and Paasche price relatives using the following standard formulas,

$$\mathcal{L}_t^{c,\text{std}} = \frac{\sum_{j\in J_{t,t-1}^c} P_{jt}Q_{j,t-1}}{\sum_{j\in J_{t,t-1}^c} P_{j,t-1}Q_{j,t-1}}, \qquad \mathcal{P}_t^{c,\text{std}} = \frac{\sum_{j\in J_{t,t-1}^c} P_{jt}Q_{jt}}{\sum_{j\in J_{t,t-1}^c} P_{j,t-1}Q_{jt}}, \tag{1}$$

where (P_{jt}, Q_{jt}) denotes price and quantity, respectively, for product j and month t. I then compute a monthly Fisher index by taking the geometric mean of the monthly Laspeyres and Paasche price relatives and chaining them together. The above formulas produce a price index for software in group c, which can be defined for any grouping of software. In the empirical analysis, I compute both a market-level and category-level price indexes.

3.2 Mudgett-Stone price index

I construct a Mudgett-Stone annual index following the description in Diewert (1998). Each product is defined both by its description and the month in which it was sold. Hence, a product sold in March of year t is compared with its namesake in March of the base year. Software with the same description but sold in different months are considered different products. The base year is set to be the previous year when constructing price relatives. Denote $\mathcal{L}_t^{c,\mathrm{MS}}$ as the Laspeyres Mudgett-Stone monthly price relatives for software in group c and $\mathcal{P}_t^{c,\mathrm{MS}}$ the Paasche. The Laspeyres and Paasche price relatives are computed using the following formulas,

$$\mathcal{L}_t^{c,\mathrm{MS}} = \frac{\sum_{j\in J_{t,t-12}^c} P_{jt}Q_{j,t-12}}{\sum_{j\in J_{t,t-12}^c} P_{j,t-12}Q_{j,t-12}}, \qquad \mathcal{P}_t^{c,\mathrm{MS}} = \frac{\sum_{j\in J_{t,t-12}^c} P_{jt}Q_{jt}}{\sum_{j\in J_{t,t-12}^c} P_{j,t-12}Q_{jt}}. \tag{2}$$

I aggregate to the annual frequency by taking a weighted average,

$$\mathcal{L}_a^{c,\mathrm{MS}} = \sum_{s=1}^{12} w_s^a \mathcal{L}_s^{c,\mathrm{MS}}, \qquad \mathcal{P}_a^{c,\mathrm{MS}} = \sum_{s=1}^{12} w_s^a \mathcal{P}_s^{c,\mathrm{MS}}. \tag{3}$$

The twelve months summed over in the above equation correspond with a calendar year a and w_s^a is the share of annual revenue for calender year a earned in month s. Finally, I compute an annual Fisher index by taking the geometric mean of the annual Laspeyres and Paasche price relatives and chaining them together. The above formulas produce a price index for software in group c, which can be defined for any grouping of software. In the empirical analysis, I compute both a market-level and category-level price indexes. While I focus on seasonality in December, this approach accounts for seasonality in each month of the year, and so the resulting index is a useful benchmark.

3.3 The Heterogeneous Price Index

In this subsection, I first lay out the motivation behind the construction of the Heterogeneous price index. I then detail the formulas behind its construction. Lastly, I describe an important issue with regard to practically implementing this price index.

3.3.1 Motivation

As detailed in Diewert (1998), the Mudgett-Stone approach is based on a representative consumer framework and so accounts for seasonality by assuming that the representative consumer's tastes change from season-to-season. In the software example, this translates into a representative consumer having different tastes in each month of the year.

1997	1998	1999	2000	2001	2002	2003	Average
0.99588	0.99362	1.01027	1.00233	1.00362	1.00641	0.99397	1.00087

Table 5: November to December Price Relatives

The Mudgett-Stone index provides an accurate measure of the change in cost-of-living under the assumption that a representative consumer can provide a good approximation of consumer behavior. The nature of the winter-holiday seasonality within the prepackaged software market, however, challenges this assumption. The overall surge in units sales in December is too large to be explained by increased shopping intensity from the same pool of households who show up throughout the year (see Figure 2). But if new households show up in December, how are these casual shoppers different from the regular shoppers who buy throughout the year? A New York Times article describes how the video game industry retailers reconfigure stores for the winter holiday to cater these casual, once-a-year shoppers.[15]

There is also empirical evidence that these December casual shoppers are different from regular shoppers. With the surge in December prepackaged software sales, a signal of high demand, we would expect an accompanying rise in price. In the data, however, there is at most a slight uptick in price. Using the original monthly data, a maximum-overlap Fisher price index computes an average price increase of only 0.09 percent in December (see Table 5). This dynamic in the data of average price *not* climbing during periods of high demand is a puzzle seen in other retail markets and is an active field of research. Given software is durable and its market is characterized by monopolistic competition, Bils (1989) is most relevant. That paper considers a monopolist selling a good to both first-time and repeat customers and shows that in periods with many new potential customers, the monopolist lowers its markup. This pricing policy generates a time-series for prices that appears to show little response to shifts in demand.[16] Given the traditions of gift-giving over the winter-holiday in the U.S., software consumers can

[15] "Casual Fans Are Driving Growth of Video Games," Seth Schiesel, *The New York Times*, September 11, 2007.

[16] Bils (1989) discusses how these results would extend to a version of the model with monopolistic competition.

be categorized into two types: regular, repeat customers and first-time, casual buyers. While regular customers buy throughout the year, casual buyers crowd into the market in December, spurred by the holiday season. According to Bils (1989), this description of consumer demand would explain the puzzling behavior of prices not rising over the winter holiday.

This characterization of consumers, however, implies that a representative framework would not provide a good approximation of consumer behavior. This is especially true for those segments of prepackaged software which experience large seasonal effects, such as Education and PC Games. Rather, a more accurate way to characterize consumer's behavior would be to separate consumers into 2 types. The first type of consumer would be regular or repeat shoppers who are in the market throughout the year. The second type of consumer only shows up in December. Importantly, this type of heterogeneity is not nested within the Mudgett-Stone framework because both types of consumers purchase products in December.

3.3.2 Method

To explicitly account for this type of consumer heterogeneity, I propose constructing separate indexes for each type of consumer. These indexes are then averaged together to produce the Heterogeneous price index. The Implementation subsection that follows details how the two types of consumers can be identified in the data. For now, take as given there are data on units sold for each type of consumer. Both consumers see the same prices in the market, but purchase different amounts. Denote $\hat{Q}^i_{jt}$ as the unit sales of product j to consumer type $i = \{1, 2\}$ in month t, where $\hat{Q}^1_{jt} + \hat{Q}^2_{jt} = Q_{jt}$. For prepackaged software, we are only concerned about the winter-holiday seasonal variation. By assumption then, when t is *not* December, $\hat{Q}^2_{jt} = 0$.

The first index measures the constant-quality price change for regular, type 1, consumers who show up throughout the year. I measure the price change faced by these consumers using a maximum-overlap matched-model approach. Let $\mathcal{L}^{c,1}_t$ be the Laspeyres monthly price relatives of sub-category c of software products for the type 1 consumer and $\mathcal{P}^{c,1}_t$ the Paasche. The construction of $\mathcal{L}^{c,1}_t$ and $\mathcal{P}^{c,1}_t$ follow the same formulas detailed in equation 1, where $\hat{Q}^1_{js}$ is substituted in place of Q_{js}. The Laspeyeres and Paasche price relatives are then chained together to produce annual price relatives,

$$\mathcal{L}_a^{c,1} = \prod_{s=1}^{12} \mathcal{L}_s^{c,1}, \qquad \mathcal{P}_a^{c,1} = \prod_{s=1}^{12} \mathcal{P}_s^{c,1}. \tag{4}$$

The second index measures the constant-quality price change for the second type of consumer who only shows up in December. I use a year-over-year approach to measure the constant-quality price change faced by these once-a-year consumers, similar to the formulas outlined in the Mudgett-Stone section above. Because the casual type 2 consumers only show up in December of each year, I construct annual Laspeyres and Paasche indexes for these consumers by looking at the change in prices in December relative to the previous December,

$$\mathcal{L}_a^{c,2} = \frac{\sum_{j \in J^c_{t,t-12}} P_{jt} \hat{Q}^2_{j,t-12}}{\sum_{j \in J^c_{t,t-12}} P_{j,t-12} \hat{Q}^2_{j,t-12}}, \qquad \mathcal{P}_a^{c,2} = \frac{\sum_{j \in J^c_{t,t-12}} P_{jt} \hat{Q}^2_{jt}}{\sum_{j \in J^c_{t,t-12}} P_{j,t-12} \hat{Q}^2_{jt}}. \tag{5}$$

After constructing the Laspeyres and Paasche indexes for the software group c for each consumer type, I then combine the Laspeyres and Paasche indexes using annual revenue weights,

$$\mathcal{L}_a^{c,H} = \mathcal{L}_a^{c,1}(1 - \omega_a^L) + \mathcal{L}_a^{c,2} \omega_a^L \tag{6}$$

$$\left[\mathcal{P}_a^{c,H}\right]^{-1} = \left[\mathcal{P}_a^{c,1}\right]^{-1} (1 - \omega_a^P) + \left[\mathcal{P}_a^{c,2}\right]^{-1} \omega_a^P \tag{7}$$

where

$$\omega_a^L = \frac{P_{a-1} \hat{Q}^2_{a-1}}{P_{a-1} Q_{a-1}}, \qquad \omega_a^P = \frac{P_a \hat{Q}^2_a}{P_a Q_a}. \tag{8}$$

Finally, I take the geometric mean of the annual Laspeyres and Paasche indexes to construct an annual Fisher price index. As evidenced from its construction, this Heterogeneous price index is an average of a month-to-month and year-over-year price index. As discussed in detail later, however, the Heterogenous price index is not simply a weighted average of the maximum-overlap Fisher and Mudgett-Stone price indexes discussed above. Accordingly, its price relatives are not bounded by the maximum-overlap Fisher and Mudgett-Stone price relatives.

3.3.3 Implementation

Because the prepackaged software data do not have demographic information, a complication with the Heterogeneous index is determining how to split the data between both types of consumers in December of each year. By construction, only the first type of consumer is shopping in months 1 through 11. Both types of consumers pay the same price for products in December (i.e. there is one market-clearing price). To split out unit sales of software between consumers, I use an ad hoc approach. I turn back to the seasonally-adjusted series created by the x-12-ARIMA software and set type 1 consumer unit sales in December equal to the seasonally-adjusted unit value. The difference between the non-seasonally-adjusted and seasonally-adjusted unit values in December is then defined as type 2 consumer sales. In section 2.2 I labeled this difference as the "seasonal" component. In essence, I are assuming that the extra bump in units sold in the fourth month is attributed to type 2 consumers. Using this approach to divide total units sold into sales to type 1 and type 2 consumers, I can use the formulas detailed above to construct the Heterogeneous price index.

In most cases, there will not be data available which will indicate sales by type of consumer. Hence, some sort of assumption will need to be made to divide total units sold among the types of consumers. My approach of assigning the seasonal component of sales to the casual type 2 consumers has at least two advantages. First, this division of sales accords well with the underlying premise behind the winter holiday seasonality. The data indicate that the burst of sales over December is driven by once-a-year shoppers, and so assigning the extra bump in sales for the month to the type 2 consumers seems reasonable. While this division of sales is ad hoc, I argue it provides a good first-cut approximation to the true division of sales between regular and once-a-year shoppers. Second, the approach of relying on x-12-ARIMA is transparent and easy to replicate. Reassuring, the paper's results are robust to different approaches to computing the seasonal and non-seasonal components of unit sales.

Year	Price Deflators			Price Indexes		
	Reg.	Mudgett-Stone	Het.	Reg.	Mudgett-Stone	Het.
1997	100	100	100	.	.	.
1998	80.0	79.1	79.0	80.0	79.1	79.0
1999	68.2	65.7	65.9	85.3	83.0	83.5
2000	57.9	52.4	54.7	84.9	79.7	83.0
2001	50.8	44.7	47.1	87.7	85.3	86.1
2002	44.6	37.8	40.8	87.7	84.5	86.7
2003	38.3	31.7	35.2	86.1	84.0	86.2
Average	.	.	.	85.3	82.6	84.1

Table 6: Prepackaged Software Fisher Price Indexes

Notes: Reg. stands for maximum overlap Fisher index and Het. stands for the Heterogeneous price index. The average price relative is the harmonized mean of annual price relatives.

4 Results

I present the results from the three price indexes at the aggregate level. To explore the different seasonal-adjustment methods, the Mudgett-Stone and Heterogeneous price indexes are compared.

4.1 Aggregate Results

As shown in Table 6, all three price indexes measure rapid declines in prepackaged software prices from 1997 to 2003. The maximum-overlap Fisher price index records that constant-quality price declined 61.7 percent from 1997 to 2003, where the average annual price decline was 14.7 percent. The Mudgett-Stone price index measures an even faster decline in prices, with an average annual price decline of 17.4 percent. The Heterogeneous price index records an intermediate price decline, with an average annual price decline of 15.9 percent over the sample.

As a group, these results point to steady and rapid declines in prepackaged software prices. Hence the first main result is that constant-quality prepackaged software prices from 1997 to 2003 declined at an average annual rate of at least 15 percent.[17] This is a

[17]The fact the maximum overlap Fisher, Mudgett-Stone and Heterogenous price indexes are in the

significant result in itself, since the BLS reports a more timid decline in price.[18] From 1998 to 2003 and using a Laspeyres index, the BLS reports an average annual price decline of 7.7 percent. Of course, the BLS uses a Laspeyres price index, while the price indexes reported here are Fisher price indexes. To control for differences in index methodology I construct a monthly Laspeyres index using the NPD scanner data without any seasonal adjustment and find it delivers an average annual price decline of 11.0 percent at an annual rate. Consequently using the NPD data results in an average constant-quality price decline that is 3.3 percentage points greater than the BLS number. The remaining 4 or 5 percentage points difference between the BLS index and those presented here are attributable to differences in index construction, especially the use of a Fisher index. Figure 5 graphs the BLS price index and illustrates the comparison to the Laspeyres and Fisher price indexes constructed using the NPD scanner data without any seasonal adjustment.

There are at least two reasons why the different data sources which might explain the 3.3 percentage point discrepancy in average price declines given by the BLS and NPD-Laspeyres price indexes. First, the NPD scanner data contains 84 percent of the retail prepackaged software market while the BLS uses a random sample of products. Second, the frequency of the price data differ. Prices are computed from monthly revenue and unit sales NPD numbers, which are based on daily transaction data and so reflect economic activity throughout the month. The BLS index uses price data that is gathered once a month.[19]

Differences in the base period is another possible explanation behind the measure of constant-quality price decline. I use the maximum-overlap method and so update the basket of goods used to construct the index every month. In contrast, the BLS only periodically updates the basket. Hence, new products are introduced into the basket with a delay, which causes the BLS to track an older set of products relative to the basket of goods used to construct this paper's price indexes. It may be these older

same ballpark, despite the large December seasonality, reflects the smooth price decline of prepackaged software over its product cycle along with the average software's product life lasting more than a year.

[18]The BLS index is the U.S. city average for Computer Software and Accessories series. The BLS only publishes a non-seasonally adjusted version of this price index.

[19]See Feenstra and Shapiro (2003) for a collection of articles concerning the promise and challenges of using scanner data to produce economic statistics.

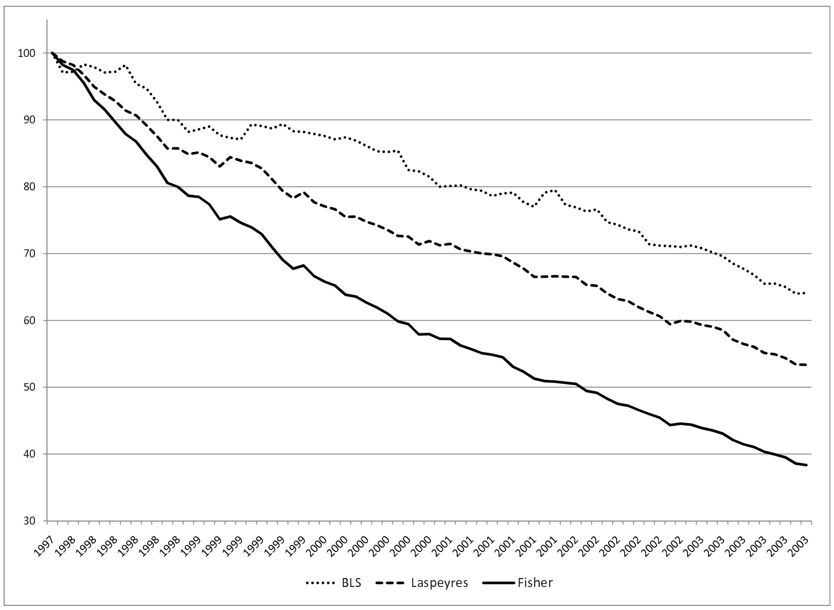

Figure 5: Comparison to BLS index

Note: The BLS index is the U.S. city average for Computer Software and Accessories and is not seasonally adjusted. The Laspeyres and Fisher price indexes were constructed from the NPD Group scanner data, also without any seasonal adjustment.

software products have slower rates of price declines, contributing to the difference in average annual price decline between the BLS consumer price index and the Laspeyres price index based on the scanner data. By construction, however, the Mudgett-Stone's year-over-year approach considers an older set of products and yet produces a measure of average price declines that is greater than the maximum-overlap approach. This suggests that differences in updating the basket of goods does not explain much of the difference between this paper's measure of price declines and those published by the BLS.

4.2 Differences in Seasonal Adjustment

Given the three price indexes, the natural question is which index provides the most accurate measure of constant-quality price change. I argue the Heterogeneous price index provides the best measure of constant-quality price change for two main reasons. First, from the data, it is clear there is substantial seasonal variation in prepackaged software unit sales in December (see Section 2.2). Hence, some adjustment for seasonal variation is necessary. Second, the source of the winter holiday sales spike is likely due to consumer heterogeneity (see Section 3). The Heterogeneous price index directly accounts for consumer heterogeneity, unlike the Mudgett-Stone price index.

While at the aggregate level there do not seem to be substantial differences between the three price indexes, large differences do appear at the disaggregated level. Because there is already a large body of empirical work which examines differences between seasonally-adjusted and non-seasonally-adjusted price indexes, here I focus on the differences between the two seasonally-adjusted price indexes: the Mudgett-Stone and Heterogeneous price indexes.[20] Large differences between these two indexes appear at the category level, as shown in Table 7. For System Utilities software, the difference between the average price relatives from the two price indexes is a hefty 11.0 percent. The categories of PC Games and Personal Productivity also have substantial differences of 3.5 and 3.2 percent respectively in the average price relatives of the two indexes. Even for categories where the average price relatives are close, significant differences crop up at the annual frequency. Education software, for example, has an average price relative

[20]For a recent and thorough overview of accounting for seasonality in price indexes, see Diewert et al (2009).

Year	Business			Education			Finance			Imaging/ Graphics		
	Reg.	MS	Het.	Reg.	MS	Het.	Reg.	MS	Het.	Reg.	MS	Het.
1998	90.1	92.6	89.7	76.9	75.9	76.3	96.3	98.1	97.4	83.3	89.4	83.5
1999	98.2	91.9	97.9	88.8	81.3	87.4	97.1	135.4	94.9	85.7	86.0	84.5
2000	96.2	91.5	98.3	82.6	82.0	80.6	95.8	85.4	81.2	86.9	81.5	85.0
2001	89.3	96.7	94.8	86.4	85.0	85.9	103.4	101.3	84.7	93.8	83.3	92.7
2002	93.8	89.5	93.3	86.0	86.4	85.0	97.2	91.6	92.2	91.1	88.1	90.6
2003	94.3	93.9	94.3	83.5	88.1	83.3	90.0	89.2	96.3	89.6	87.2	88.8
Average	93.6	92.6	94.6	83.8	82.9	82.9	96.5	97.9	90.7	88.3	85.8	87.4

Year	Operating System			PC Games			Personal Productivity			System Utilities		
	Reg.	MS	Het.	Reg.	MS	Het.	Reg.	MS	Het.	Reg.	MS	Het.
1998	90.5	94.6	95.3	68.8	63.9	68.2	83.1	82.4	81.8	76.0	73.2	77.1
1999	100.4	97.3	99.7	75.4	67.8	74.2	83.0	76.7	82.5	73.9	62.0	72.7
2000	98.2	97.5	98.1	75.8	70.1	74.7	80.7	82.6	80.0	99.2	70.1	99.0
2001	100.3	102.1	100.4	78.9	75.9	78.4	88.4	82.5	88.0	95.5	85.7	95.1
2002	96.4	98.7	96.6	78.8	77.3	78.3	85.6	85.0	86.0	95.5	90.1	95.6
2003	100.3	99.0	100.4	75.2	72.3	73.5	89.5	79.4	89.6	94.9	88.6	94.8
Average	97.6	98.1	98.4	75.3	70.9	74.4	84.9	81.3	84.5	87.9	76.8	87.8

Table 7: Price Indexes by Category

Notes: Reg. stands for maximum overlap Fisher index, MS stands for Mudgett-Stone index and Het. stands for the Heterogeneous index. The average price relative is the harmonized mean of annual price relatives.

of 82.9 under both the Mudgett-Stone and Heterogeneous methods. But in 2003 there is a large 4.8 percentage point difference between these two indexes.

As laid out in Section 3, there are fair number of differences in the construction of the Mudgett-Stone and Heterogeneous price indexes which contribute to the discrepancies in the measured constant-quality price change. Recall however, that the Heterogenous price index is an average of a month-to-month and a year-over-year index. The weight of the year-over-year index is determined by the amount of seasonality observed in the data. For products with no seasonality, the Heterogenous price index simplifies to the maximum overlap Fisher price index. For example, when looking at System Utilities, a category with little seasonality, the Heterogenous and regular maximum overlap price indexes are quite close.[21] In contrast, the Mudgett-Stone price index measures a much more rapid decline in prices. This difference is most likely driven by the small sample behind the Mudgett-Stone index. As shown in the last column of table 8, for System Utilities, the Mudgett-Stone price index is using only 8 percent of the total observations.

Even though the Heterogenous price index is an average of a month-to-month and year-over-year index, it is not bounded by the maximum-overlap and Mudgett-Stone price indexes (e.g. see the price indexes for Business software in 2000). This is because the construction of the Heterogenous price index involves splitting unit sales between two types of consumers. Hence the distribution of revenue weights across price relatives will likely quite different for the once-a-year shoppers in the Heterogenous index compared to the representative consumer in the Mudgett-Stone index. Similarly, the the distribution of revenue weights across price relatives will differ for the regular shoppers in the Heterogenous index compared to the representative consumer in the maximum-overlap index.

The results above show that while the Mudgett-Stone and Heterogenous price indexes produce aggregate average price declines for prepackage software that are in the same ballpark, there are significant differences between the two indexes for particular years or at the disaggregate software category level. Given the Heterogenous price index produces different measures of constant-quality price change, it is important to assess

[21] For Business software, which also typically exhibits little seasonality, the Heterogenous and regular maximum overlap price indexes are also close, except for 2001. In that year, there was some unusual seasonality in Business software.

Category	Maximum overlap		Mudgett-Stone	
	unweighted	weighted	unweighted	weighted
Business	72	95	42	39
Education	85	97	59	34
Finance	80	94	48	26
Imaging/Graphics	79	97	50	30
Operating System	71	90	42	42
PC Games	87	90	61	23
Personal Productivity	83	97	56	29
System Utilities	72	97	38	8
All	82	93	55	27

Table 8: Percent of Matching Observations

Notes: A cell entry in the unweighted column is the mean percent of observations which are matched to products in the base period. For the weighted column, a cell entry is the revenue-weighted mean percent of matched observations.

the advantages to using this proposed price index vis-a-vis the Mudgett-Stone approach. Like the Mudgett-Stone index, the Heterogenous price index accurately captures the once-a-year arrival of casual consumers in December using a year-over-year approach. Unlike the Mudgett-Stone index, however, the Heterogenous index only constructs the year-over-year index for products with seasonality and assigns revenue weights based on the observed seasonal component. Hence, unlike the Mudgett-Stone index, software with no December seasonality will *not* be part of a year-over-year index.[22] This distinction matters since a main drawback with a year-over-year index is the potential for a small, unrepresentative sample because of product entry and exit. Table 8 shows the percent of matching observations for the Mudgett-Stone index by prepackaged software category. When using revenue weights, this percent is always below 50 percent for the Mudgett-Stone index and even reaches a low of 8 percent. The Heterogenous price index avoids this problem by also using a month-to-month index. This index, representing price changes for the regular software shoppers, uses most of the data (see the first two columns of table 8).

[22]Further, for software with minimal seasonality in December, the year-over-year sub-index for the Heterogenous approach will receive a small weight.

Given the Heterogenous index is a weighted average of a year-over-year and month-to-month index, it offers the same transparency and practicality as the Mudgett-Stone and maximum-overlap Fisher indexes. The main disadvantage of the Heterogenous index, however, is determining how to split unit sales between the regular and casual shoppers. For prepackaged software, I only focused on the December seasonality and used an x12-ARIMA algorithm to construct unit sales series for each type of consumer. Ideally, one would be able to find survey data on the percent of sales made by each type of consumer. Reassuringly, the results presented here are robust to alternative approaches to creating a seasonally-adjusted unit sales series (i.e. creating unit sales series for each type of consumer).

5 Conclusion

In this paper, I examine the prepackaged software market using a detailed and comprehensive data set. Using three different price indexes, I find that constant-quality prices decline at an average annual rate of at least 15 percent in the sample. This is a substantially greater fall in price than reported by the BLS. Second, I consider the seasonal variation in the prepackaged software market. I find that most software products experience a December sales surge, which I claim is driven by consumer heterogeneity. Specifically, the data suggests that casual, once-a-year shoppers enter the software market over the winter holiday season. Significantly, the Mudgett-Stone price index does not properly account for this type of heterogeneity. Hence, I propose a novel approach to constructing a price index that directly accounts for heterogeneity. This approach entails constructing separate price indexes for each type of consumer, and then averaging the price indexes. This Heterogeneous price index measures an average annual price decline of 15.9 percent for prepackaged software. Properly accounting for this heterogeneity is important, because the Mudgett-Stone and Heterogeneous price index produce different estimates of constant-quality annual price change.

More broadly, this research suggests that real consumer expenditures on software may be understated in the national accounts. This is because the BLS's consumer price index for computer software, which the BEA uses to deflate nominal personal consumption expenditures on software, measures a markedly smaller decline in software prices compared to the Heterogeneous index constructed using the NPD Group scanner data. The national accounts then, may not fully reflect the growth rate of real personal consumption expenditures on prepackaged software. Further research should be done on measuring constant-quality price change for software, to ascertain whether the BLS price index understates the decline of prepackaged software prices and, if so, by how much.

An issue touched upon, but outside the scope of this paper, is the endogenous entry of new products around the beginning of the winter-holiday season. I ignore these new goods with the matched-model approach. But the entry of new goods at the beginning of the fourth month is closely related to the seasonality issues discussed in this paper. Untangling these two forces, however, likely requires a formal and sophisticated model of firm and consumer behavior, a promising avenue for future research.

References

[1] Abel, J., Berndt, E., White, A. 2003, "Price Indexes for Microsoft's Personal Computer Software Products", NBER working paper 9966.

[2] Aizcorbe, A., Bridgman, B., Nalewaik, J. 2010. "Heterogeneous car buyers: A stylized fact", emphEconomics Letters, vol 109, pages 50-53.

[3] Aizcorbe, A., Copeland, A. 2006. "Intermittent Purchases and Welfare-Based Price Deflators for Durable Goods", manuscript, Bureau of Economic Analysis working paper 2007-02.

[4] Alterman W., Diewert. W.E., Feenstra, R.C. 1999. *International Trade Price Indexes and Seasonal Commodities*, Washington D.C., U.S. Department of Labor.

[5] Bils, M. 1989. "Pricing in a Customer Market", *Quarterly Jounral of Economics*, vol 104(4), pages 699-718.

[6] Diewert, W.E., 1998. "High Inflation, Seasonal Commodities and Annual Index Numbers", *Macroeconomic Dynamics*, vol 2, pages 456-471.

[7] Diewert, W.E. 1999. "Index Number Approaches to Seasonal Adjustment", *Macroeconomic Dynamics*, vol 3, pages 48-68.

[8] Diewert W.E., Balk, B.M., Fixler, D., Fox, K.J., Nakamura, A.O. 2009. *Price and Productivity Measurement: Volume 2 – Seasonality*, Trafford Press.

[9] Feenstra R., Shapiro, M. 2003 *Scanner Data and Price Indexes* Chicago: NBER and University of Chicago Press.

[10] Fisher, F.M., Griliches, Z. 1995. "Aggregate Price Indices, New Goods, and Generics", *Quarterly Journal of Economics*, vol 110, pages 229-244.

[11] Gandal, N. 1994. "Hedonic Price Indexes for Spreadsheets and an Empirical Test of Network Externalities", *RAND journal of Economics*, vol. 24, pages 160-170.

[12] Griliches, Z., Cockburn, I. 1994. "Generics and New Goods in Pharmaceutical Price Indexes", *American Economic Review*, vol 84., pages 1213-1232.

[13] Jorgenson, D.W. 2001. "Information Technology and the U.S. Economy," *American Economic Review*, vol 91, pages 1-32.

[14] McCahill, R.J. 1997, "A Quality-Adjusted Price Index for Prepackaged Software", Virginia Polytechnic Institute and State University Master's Thesis.

[15] Nesmith, T.D. 2007. "Rational Seasonality", In William A. Barnett and Apostolos Serletis, eds., *Functional Structure Inference* United Kingdom: Elsevier, 2007.

[16] Oliner, S., Sichel, D. 1994. "Computers and Output Growth Revisited: How Big is the Puzzle?", *Brookings Papers on Economic Activity*, vol 2, pages 273-317.

[17] Parker, R., Grimm, B. 2000. "Recognition of Business and Government Expenditures for Software as Investment: Methodology and Quantitative Impacts: 1959-98", manuscript, Bureau of Economic Analysis.

[18] Prud'homme, M., Sanga, D., and Yu, K. 2005. "A Computer Software Price Index Using Scanner Data", *Canadian Journal of Economics*, vol. 38, pages 999-1017.

Month in the Product Cycle	Price		Sales	
	coefficient	standard error	coefficient	standard error
2	-0.015	0.00298	0.192	0.00697
3	-0.037	0.00299	0.217	0.00670
4	-0.046	0.00301	0.182	0.00705
5	-0.068	0.00313	-0.067	0.00732
6	-0.088	0.00324	-0.189	0.00758
7	-0.102	0.00328	-0.160	0.00767
8	-0.129	0.00339	-0.254	0.00793
9	-0.141	0.00350	-0.379	0.00819
10	-0.155	0.00350	-0.342	0.00820
11	-0.141	0.00362	-0.468	0.00848
12	-0.191	0.00377	-0.534	0.00882
13	-0.220	0.00384	-0.555	0.00898
14	-0.284	0.00401	-0.567	0.00938
15	-0.296	0.00418	-0.560	0.00978
16+	-0.359	0.00297	-0.826	0.00695

Table 9: Regression Coefficient Estimates and Standard Errors

Notes: Cell entries are the estimated coefficients and associated standard errors from a regression of the log of prices (sales) on product cycle dummies, with fixed effects for software products. The dummy variable 16+ is equal to 1 for months 16 and greater in the product cycle.

A Appendix

Table 9 reports the results from the regressions of the log of sales (prices) on product cycle dummies, with fixed effects for each software product and using revenue weights. The coefficients were plotted in figure 1.