# Opportunities and Challenges: Lessons from Analyzing Terabytes of Scanner Data

Serena Ng*

Columbia University

December 7, 2016

**Abstract**

This paper seeks to better understand what makes big data analysis different, what we can and cannot do with existing econometric tools, and what issues need to be dealt with in order to work with the data efficiently. As a case study, I set out to extract any business cycle information that might exist in four terabytes of weekly scanner data. The main challenge is to handle the volume, variety, and characteristics of the data within the constraints of our computing environment. Scalable and efficient algorithms are available to ease the computation burden, but they often have unknown statistical properties and are not designed for the purpose of efficient estimation or optimal inference. As well, economic data have unique characteristics that generic algorithms may not accommodate. There is a need for computationally efficient econometric methods as big data is likely here to stay.

Keywords: big data, random sub-sampling, leverage score sampling, seasonal adjustment
JEL Classification: C1, C4, E3

[*] Department of Economics, 420 W 118 St, New York, NY 10027. Email: serena.ng at columbia.edu

Financial support from the National Science Foundation (SES-0962431) is gratefully acknowledged. I would like to thank David Weinstein for introducing me to this work and Jessie Handbury for getting me started. I also thank Jushan Bai, Christos Boutsidis, Jean-Jacques Forneron, Matt Shum, and Juan Ospina for helpful discussions. This work would not be possible without the contribution of Rishab Guha and Evan Munro. I am deeply indebted to their help. All errors are mine.

## 1 Introduction

The goal of a researcher is often to extract signals from the data, and without data, no theory can be validated or falsified. Fortunately, we live in a digital age that has an abundance of data. According to the website Wikibon (`www.wikibon.org`), there are some 2.7 zetabytes data in the digital universe.[1] The U.S. Library of Congress collected 235 terabytes of data as of 2011. Facebook alone stores and analyzes over 30 petabytes of user-generated data. Google processed 20 petabytes of data daily back in 2008, and undoubtedly much more are being processed now. Walmart handles more than one million customer transactions per hour. Data from financial markets are available at ticks of a second. We now have biometrics data on finger prints, handwriting, medical images, and last but not least, genes. The 1000 Genomes project stored 464 terabytes of data in 2013 and the size of the database is still growing.[2] Even if these numbers are a bit off, there is lot of information out there to be mined. The data can potentially lead economists to a better understanding of consumer and firm behavior, as well as the design and functioning of markets. The data can also potentially improve the monitoring of traffic control, climatic change, terror threats, causes and treatment of health conditions. It is not surprising that many businesses and academics are in a big data frenzy. The Obama Administration announced the Big Data Research and Development Initiative in 2012.[3] The National Bureau of Economic Research offered lectures on big data in two of the last three summer instiutes. Courses on big data analysis often have long waiting lists.

Many economists have written about the potential uses of big data. New overview articles seem to appear in REPEC every month. Some concentrate on the economic issues that can be studied with the data, as in the excellent articles by Einav and Levin (2013, 2014), Athey (2013). Other surveys take a more statistical perspective. For example, Varian (2014) considers machine learning tools that are increasingly popular in predictive modeling. Fan et al. (2014) warns about the possibility of spurious correlation, incidental endogeneity, and noise accumulation that come with big data and suggests new methods to handle these challenges. While the use of big data in predictive analysis has drawn the most attention, much of economic analysis is about making causal statements. Belloni et al. (2014) discusses how regularization can permit quality inference about model parameters in high-dimensional models. Athey and Imbens (2015) uses machine-learning methods to estimate treatment effects that may differ across subsets of the population.

As with these studies, I also consider methods specific to big data. But instead of predictive modeling and taking the data as given, I focus on data preprocessing, perhaps the most time consuming step of a big data analysis. This paper was initially motivated by the curiosity to learn what makes big data analysis different, how far can our existing econometric tools take us, and getting a sense of what issues need to be addressed if big data are here to stay, using as case study

[1] 1024 Megabyte = 1 Gigabyte, 1024 Gigabytes = 1 Terabytes, 1024 Terabyte=1 Petabyte, 1024 Petabyte=1 Exabyte, and 1024 Exabyte=1 Zetabyte.

[2] The project seeks to find most genetic variants that have frequencies of at least 1% in the population.

[3] See `http://www.whitehouse.gov/blog/2012/03/29/big-data-big-deal`.

four terabytes of weekly retail sales data collected between 2006 and 2010. A distinctive feature of the dataset is that it has direct measures of prices and quantities. I use the opportunity to analyze the cyclical aspects of the quantity data. This is interesting because the Great Recession of 2008 is in this sample, and official consumption data do not come at higher than a monthly frequency. The project gives me a better understanding of the limitations of statistics/econometrics in big data analysis, and why methods in the domain of data science are useful.

A gigabyte of data can be easily analyzed on our desktop computers using our favorite statistical software packages. The problem is that methods which we understand and work well with small datasets may not be big data friendly or scalable. Even though I have four terabytes of data, it is impossible to analyze them all at once. The memory requirement is beyond the capacity of our computers even with unlimited financial resources. Aggregation, whether in the time, spatial, or product dimension, would seem to take away features that make the data special. Fortunately, even if we could analyze all the data, it might not be necessary to do so. Studying a subset of the data might suffice, provided that the subset is appropriately assembled. Hence the first part of this paper explores two random subsapling algorithms developed by computer scientists to accurately approximate large matrices. Random subsampling is neither efficient nor necessary when the sample size is manageable. In a big data setting, random sampling not only speeds up the analysis; it is a way to overcome the constraints imposed by the computing environment. However, the subspace-sampling algorithms considered are developed to run fast and have desirable 'worse case error bound', quite distinct from the optimality criteria such as mean-squared error and consistency that we typically use. There is thus a need to evaluate these algorithms in terms of quantities that we analyze. This is difficult when the probability structure of the data is not specified.

Business cycle analyses typically use data collected by government agencies that also handle the data irregularities. With the Nielsen data, the task of removing seasonal effects is left to the user. The challenge is that weekly seasonal variations are not exactly periodic. Structural modeling on a series by series basis may deliver a filtered series that is statistically optimal, but this is impractical when we have millions if not billions of highly heterogeneous series to analyze. Hence the second part of this paper explores a practical approach to modeling the seasonal effects and have mixed success. I find that removing the seasonal effects at individual level is no guarantee that the seasonal variations at the aggregate level will be removed. The exercise does, however, suggest promising ways of handling seasonality that need to be further explored.

More generally, the volume, heterogeneity, and high sampling frequency that generate excitement about the data are precisely what make extracting signal from the data difficult. Big data creates a need for econometric methods that are easy to use, computationally fast, and can be applied to data with diverse features. Accomplishing these objectives may entail a change from the current practice of customizing a model to a particular data type. The difference is a bit like shopping at a general merchandise store versus a specialty store; there is a tradeoff between quality and convenience. The non-probabilistic methods developed by data scientists enable efficient computations, but they are not developed with estimation and inference in mind. It is an open question whether computation efficiency and statistical efficiency are compatible goals. It is also debatable if precision of estimates obtained in a data rich environment can be judged the same way as when the sample size is small. Understanding the statistical underpinnings of computationally efficient methods can go a long way in easing the transition to big data modeling. This can be important as big data are likely here to stay.

# 2 Data Analysis in the Digital Age

This section has two parts. Subsection 1 draws attention to the challenges that big data pose for traditional statistical modeling that is also the foundation of econometrics. Subsection 2 highlights some characteristics of big data and summarizes features of the Nielsen scanner data used in the analysis of Section 3 and 4.

## 2.1 Data Science and Statistics

A lot has been written about 'big data' in recent years, but not everyone has the same notion of what big data is. Labor and health economists have long been analyzing big surveys, census data and administrative records such as maintained by the Social Security Administration, Medicare and Medicaid. Increasingly, macroeconomists also turn to big data to study the price determination process, sometimes involving unpublished data. But once access to the data is granted, analysis of these pre-Google big data can proceed using existing hardware and software packages like STATA and MATLAB.

The post-Goggle data that concern this study are the large and complex datasets that are not collected through surveys, not supported by official agencies, and cannot be stored or analyzed without switching to a new computing environment at some point. If 8 bytes (64 bits) are used to store a real number, a few billion of observations for several variables would be beyond the capacity of most off-the-shelf desktop computers. What makes big data analysis different is not just that the sheer size of the dataset makes number crunching computationally challenging,[4] but also that the observations are often irregularly spaced and unstructured. Indeed, it is quite common to use three-Vs to characterize big data: large in **V**olume, come in a **V**ariety of sources and formats, and arrive at a fast **V**elocity. Some add variability and veracity to the list because the data are sometimes inconsistent in some dimensions and possibly inaccurate. Conventional methods designed to process data with rectangular structures often do not apply. There is no statistical agency to oversee confidentiality and integrity of the data, and the tasks of cleaning and handling the data are in the hands of researchers, many of whom have limited knowledge about database

[4] A problem is class P if it runs in polynomial time (eg. linear, quadratic, or logarithmic in the size of the input, say $n$). A problem is in NP class if its solution cannot be verified in polynomial time. An NP-hard problem is at least as hard as the hardest NP problem.

management. Python and R seem to be commonly used to prepare the data for analysis but often, programs written for one dataset are of little use for another because each dataset typically has its own quirky features.

Each of the three Vs pose interesting challenges for statistical analysis because they violate assumptions underlying methods developed for conventional data. Because of variety, it may be difficult to justify a common data generating process. Because of volume, thinking about how to conduct optimal estimation and inference is not realistic when we struggle just to find ways to summarize the massive amount of information. It would also not be useful to have complex models that cannot be estimated, or MCMC methods that cannot be completed within a reasonable time frame. Bayesian estimation would essentially be likelihood based when sample information dominates the prior. Because of velocity and volume, the standard error of estimates will be tiny. But because the noise level in big data can be high, the assumption that information increases with sample size may be questionable, an issue noted in Granger (1988). A new way of doing asymptotic analysis may well be warranted.

A big data project typically uses methods that are part statistics, part computer science, and part mathematics, and is often associated with the field of *data science*. Cleveland (2001) proposes to expand the areas of technical work in statistics and to call the new field 'data science'. Wikipedia defines the field as 'extraction of knowledge or insights from large volumes of data', thereby directly linking data science with big data. Another characterization is well summarized by how The Journal of Data Science defines its scope:- 'everything to do with data: collecting, analyzing, modeling, ..., yet the most important part is its application'. The emphasis here is the ability to apply what is learned from the data analysis to practical use, such as business analytics and predictions. In a sense, this view treats data analysis as an immediate input of production; what ultimately matters is the value of the final good.

In an influential paper, Brieman (2001) distinguishes data science from traditional statistical analysis as follows. A statistician assumes a model, or a data generating process, to make sense of the data. Econometric analysis largely follows this stochastic model paradigm. The theoretical results are not always well communicated to practitioners and not always taken to the next level after publication of the article. Brieman (2001) argues that the commitment to stochastic models has handicapped statisticians from addressing problems of interest and encourage the adoption of a more diverse set of tools. A data scientist accepts the possibility that the assumptions underlying models may not be correct. He/she therefore uses algorithms, or machine-learning methods, to map data to objects of interest, leaving unspecified the data generating process that nature assigns. Probability models and likelihoods are replaced by random forests, regression trees, boosting, and support vector machines. One looks for clusters and frequent-items in the data. The work of a data scientist often has immediate downstream uses (such as for business decisions or in gene mapping).

Big data provides a momentum boost to move away from stochastic modeling as the more data with the three V features we have, the more difficult it is to defend a model that is generally valid.

The American Statistical Association (ASA) has a working group to study the future direction of the discipline at large. The group sees collaboration with data scientists as a way for statisticians to contribute to exciting problems of the digital generation.[5] The Institute of Mathematical Statistics also recognizes the challenge that data science poses. In her 2014 presidential address, Bin Yu remarked that data science represents an inevitable (re)-merging of computational and statistical thinking. She suggests to call themselves (ie. statisticians) data scientists in order to fortify their position in the new era of data analysis, echoing the suggestion of the statistician Jeff Wu made at an inagural lecture at the University of Michigan in 1997.[6]

While statisticians are open to the idea that computer science and mathematics will play an important role in statistical analysis in the future, economists are slower to react. Most of us have little experience with big data and know little about the computational aspect of data analysis. As will be discussed in Section 3, we may well have to become active in this area of research as we are increasingly presented with opportunities to analyze big economic data, and see that there are data issues that require our knowledge and input.

## 2.2 Data Types

Most post-Google big datasets are not intentionally collected, hence they are cheap to produce compared to data produced by surveys. The big datasets used in economic analysis are usually in one of two forms. The first is generated by search engines and social media websites such as Google, Facebook, and Twitter. It is no secret that online clicks have been used to target products to potential buyers. Social media data are now more effective than data from loyalty programs in predicting repeated purchases. But web search data have many uses other than advertising, the most famous of which is probably the initial success of prediction of flu outbreaks by Ginsberg et al. (2009). A creative use of social media data is the U-report, a UNICEF project that collects text-messages from young people in Uganda. IBM researchers were able to apply machine learning methods to the tweets to learn about economic, political, and health conditions, and to alert health officials of ebola outbreaks.[7] Projects of this type are now expanded to other parts of Africa.

A second type of data comes from web searches. Such data provide information about intent and potential actions, hence can be useful for prediction. Choi and Varian (2012) finds that a small number of Google search queries can 'nowcast' car sales and shows how proxies for consumer confidence can be constructed from Google Trends data. Preis et al. (2013) computes a Future Orientation index and finds a correlation between online searches and realized economic outcomes.[8]

[5] See http://www.amstat.org/policy/pdfs/BigDataStatisticsJune2014.pdf.

[6] See also http://bulletin.imstat.org/2014/09/data-science-how-is-it-different-to-statistics/, http://magazine.amstat.org/blog/2010/09/01/statrevolution/, http://statweb.stanford.edu/~tibs/stat315a/glossary.pdf. for differences between the two fields. http://bulletin.imstat.org/2014/10/ims-presidential-address-let-us-own-data-science/

[7] http://www.research.ibm.com/articles/textual-analysis-u-report.shtml.

[8] The Future orientation index is the ratio of the volume of searchers of the future (ie 2011) to the past (ie.2010).

Koop and Onorante (2013) uses Google search data to improve short-term forecasts. Antenucci et al. (2014) uses twitter data to produce an index that can predict job loss.

A different type of big data is action-based, arising from the real-time purchases at stores such Walmart, and from charges processed by, for example, Mastercard. These databases are relatively structured and often have a business value. As an example, Target was reported to form prediction indicators from buying habits of customers going through life changing events, such as divorce and giving birth, and push to them the promotional flyers.[9]. Based on Matercard transactions, SpendingPulse$^{TM}$ claimed that its near-real time purchase data can predict spending weeks if not months ahead of other sources.

Data on prices are of particular interest to economists. The Billion Prices project gives real time inflation predictions by aggregating information in five million items sold by about 300 online retailers around the world. Handbury et al. (2013) uses a Japanese dataset with five billion observations on price and quantity to construct an ideal (Tornqvist) price index. The authors report a non-trivial difference between their measure and the official measure of inflation. This type of data is valuable when credibility of the official data is in question, as in the case of inflation in Argentina. It is also useful when release of the data is disrupted by unanticipated circumstances, such as in the case of earthquakes in Chile and Japan, see Cavallo (2012) and Cavallo et al. (2013).

## 2.3 The Nielsen Data

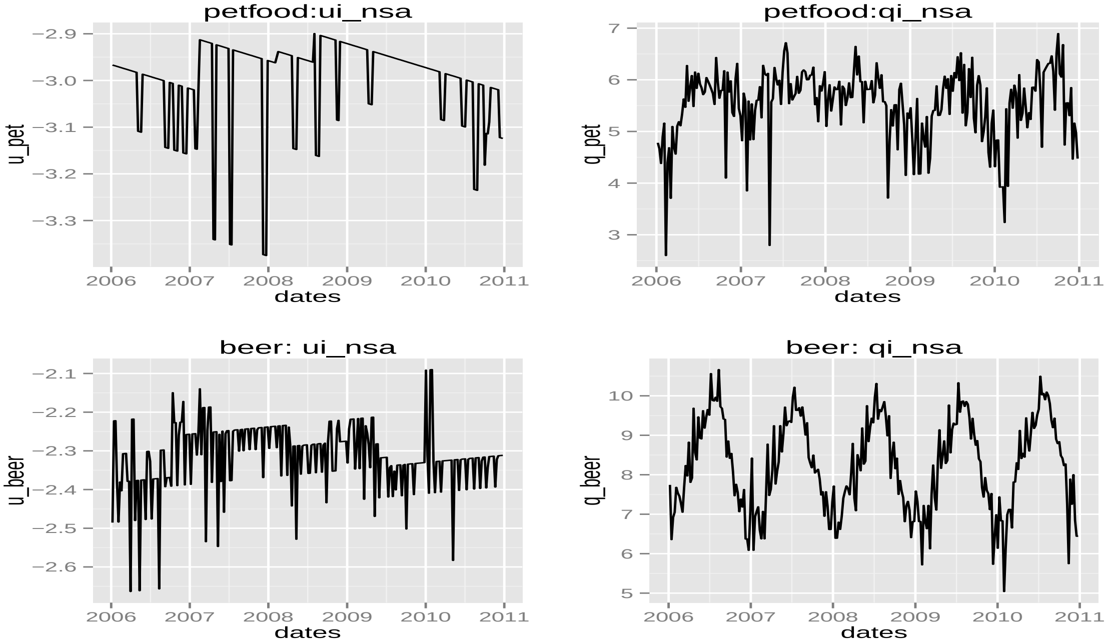
The dataset that motivates this analysis is the Retail Scanner Data collected weekly by the Nielsen marketing group. The database is managed by the Kilts center for marketing at the University of Chicago. Through a university license, the data are made available for analysis a couple of years after the actual transactions. The data are collected at 35,000 participating grocery, drug stores, and mass merchandiser affiliated with about 90 participating retail chains across 55 MSA in the U.S.. Our data are from 2006 to 2010. The dataset covers 3 million unique UPC for 1073 products in 106 product groups which are in turn classified into 10 categories: *dry groceries, frozen, dairy, deli, meat, fresh food, non-food, alcoholic beverage, general merchandise*, and *health and beauty.* The data are structured (ie. in numeric format only, audio and video files are not involved) but highly heterogeneous. There is also information about location (zip and fips county codes) and the retailer code, but retailers are not identified by name. Household level information is in a companion Nielsen Homescan Consumer Panel database which is not used in this study. The Nielsen data have been widely studied in marketing analysis of specific products.[10]

The variables of interest are units of packages sold and the price of the package, from which unit price is constructed. Several features make the data interesting to economists. First, prices and quantities are separately observed. In contrast, conventional price deflators are inferred from observations on value and quantities. Furthermore, this data are recorded at a higher frequency

[9] Tolentino (2013) analyzes loyalty programs, Goel (8-2-2014) on Facebook, and Duhigg (2-16-2012) on Target.

[10] Research papers using the data can be found in `http://research.chicagobooth.edu/nielsen/research/`

Figure 1: $u_{it}$ and $q_{it}$ (NSA): Pet Food and Beer

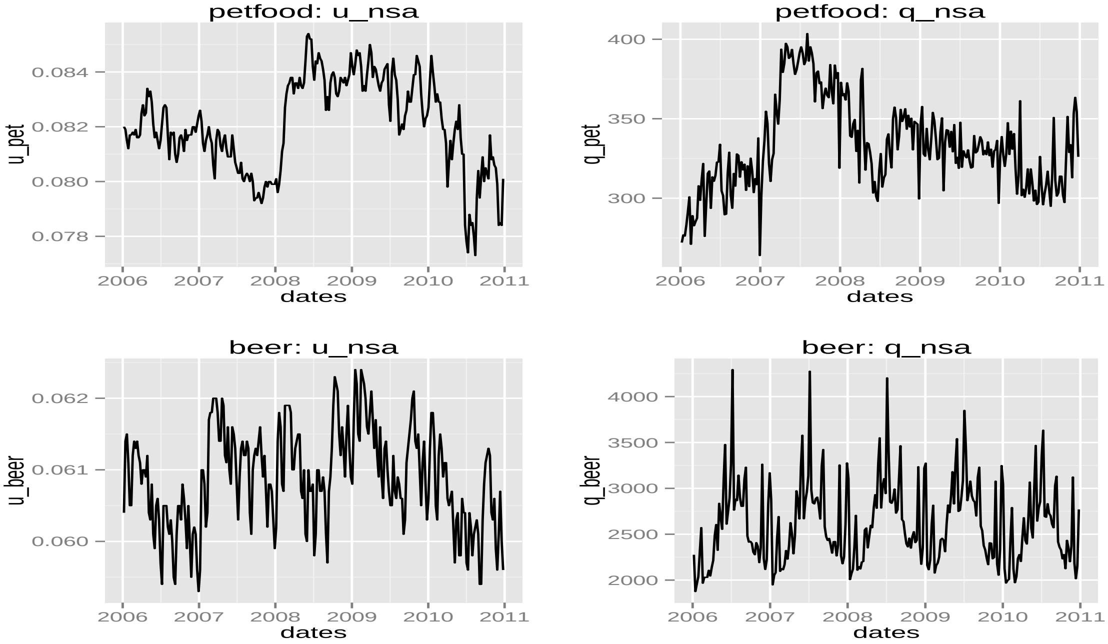


and at more locations than the official data on retail sales. In fact, few economic indicators (on price or quantity) are available at a weekly frequency. Even at a monthly frequency, there is little data available at a local level. However, the Nielsen data also has several drawbacks. The data only cover grocery store purchases and ignore services and durables which tend to be more cyclical. Furthermore, the data are not seasonally adjusted.

An increasing number of researchers are using scanner data to answer interesting economic questions. Broda et al. (2009) concludes from analyzing the Homescan data that the poor indeed pays less for food purchases, not more, as poverty analyses based on the CPI suggest. Beraja et al. (2015) constructs monthly price indexes to study the impact of local versus aggregate shocks. The indexes are constructed from bottom-up (group by group), keeping memory usage is kept at a manageable level. Coibion et al. (2015) uses an IRI database that is similar to the Nielsen data but with fewer products to study the cyclicality of sales. They aggregate the data to monthly frequency and pool the data across markets to run fixed effect regressions. Cha et al. (2015) aggregates the weekly Homescan data to annual level and finds that food consumed at home is countercyclical.

Far fewer studies have looked at the price data at the native (weekly) frequency. Even harder to find are studies that analyze the quantity data. One reason could be that there are not many predictors available at a weekly frequency for a structural demand analysis. Even at a descriptive level, analysis of the quantity data at the weekly level requires separating the cyclical from the seasonal components. I hope to make some progress on this front, given the unique opportunity of having the financial crisis in the sample of 2006-2010.

Figure 2: $\overline{u}_t$ and $\overline{q}_t$ (NSA): Pet Food and Beer



A total of six products will be analyzed: beer, foreign wine, meat, eggs, pet food and baby food. Results for lightbeer and beer, domestic wine and foreign-wine are similar and not reported. For a given product, let unit price at week $t$ be $u_{ti}$ and units sold be $q_{ti}$, where $i$ is a unique store-UPC pair. For example, Coke-zero and Coca-Cola-light sold at the same store are two different units, as are Coke-zero sold at say, Seven-Eleven and Wawa. To get an idea of features in the data, Figure 1 shows $u_{ti}$ and $q_{ti}$ for one $i$ selected from the pet food and one $i$ selected from the beer group. Figure 2 shows the unweighted mean over all $i$ in the balanced panel (denoted $\overline{u}_t$ and $\overline{q}_t$). The $u_{ti}$ series for both products are non-smooth, reflecting infrequent price changes. The downward spikes are likely due to discounts. Chevalier et al. (2003) finds evidence of price discounts around seasonal peaks in demand. The seasonal variations in the quantity data for beer are strong at both the individual and aggregate levels.

My goal is to extract the cyclical information in the $q_{ti}$ data. After linearly detrending the data, the first two principal components explain around 15 percent of the variations, suggesting the presence of pervasive variations. In the next two sections, I consider two problems encountered. The first is memory constraint which leads to investigation of random sampling algorithms. The second relates to the goal of extracting the cyclical component, which calls for the need to seasonally adjust the weekly data on a large scale. As we will see, knowledge of tools unfamiliar to economists can go some way in making the analysis more efficient, but many issues remain to be solved.

# 3 Balancing and Sketching the Data

The time it takes to perform a task on the computer depends not just on how efficient is the program written, and in what language, but also on the hardware which big data put to a serious challenge. Specifically, the computation speed depends on how frequently the data are brought from physical storage (the hard disk) to RAM, how fast and how much data can be moved from RAM to the CPU, and the queuing time which depends on the amount of the requested RAM. We have almost four terabytes of data and processing them requires a lot of RAM! The original intent was to perform all computations on a cloud server such as Amazon Web Service. Unfortunately, the user agreement restricts data storage to university owned hardware. It took months to find a feasible and efficient alternative. Eventually, my computing environment consists of a (not very fast) server that allows each job to use up to 256GB of RAM, and a desktop (2011 vintage iMac) upgraded to 24GB of RAM.

To reduce the volume of data in a systematic manner, my student helpers (Rishab Guha, Evan Munro) and I started by constructing a balanced panel for each of the products considered. This is itself a RAM and time intensive exercise. We are familiar with MATLAB and somewhat familiar with R but have no prior experience with database management or packages like PANDAS, which we subsequently use. We initially wrote programs in STATA, PYTHON and R for the same task as a way to check bugs but settled on using PYTHON. We experimented with several ways of balancing the panel. The first method keeps only those UPC-stores that are available for every week in the year and then concatenate the five years of data to keep only those UPC-stores that are available for each of the 260 weeks. The second method stacks all 260 weeks of data and selects those store-UPCs with recorded sales in every week. Eventually, we (i) manually sort the data frame by upc and store code, (ii) loop through the underlying array while keeping track of the number of observations for each unique upc/store code combination, (iii) keep only those with 260 weeks of observations. At least for python, this procedure is much faster than using the built in 'group-by' functions; runtime was cut by a factor of 20, taking about an hour to balance 71GB of data. Further tweaking and making use of the just-in-time compiler from the Numba package further reduced the runtime to about 18 minutes, making it feasible to clean all 4TB of data, should we choose to do so. The code for cleaning the data for a particular product is 33 lines. Each job uses between 144GM and 168GB of RAM depending on the size of the data for that product.

Working with balanced panels comes at the cost of incurring selection bias, as mostly likely, the smaller stores are being discarded. Eventually, this issue needs to be explored. But the analysis is now more manageable. As an example, the raw data for beer is 20.3 GB, but the balanced panel is just over 2 GB, with 15 million data points for each of the three variables price, quantity, and value of sales. Together with data on location and other store specific information, there are still over 100 million data points on beer to analyze. While this can be even be done on a desktop, having any software to read in millions of observations can be quite slow, especially when this process has to be repeated many times until the program is bug-free. But it does not seem necessary to use all the data at the debugging stage. This leads me to consider working with subsamples that preserve characteristics of the original data.

We use simple moments such as mean and variance to describe the data, principal components to highlight the dominant variations, regressions for predictions or to study the structural relations amongst variables. But how much data do we really need? The problem of efficiently analyzing a large volume of data within the constraints imposed by our software and hardware is not new. Deaton and Ng (1998) considers non-parametric regressions when the number of calculations is proportional to $NK^2$ where $N$ is the number of cross-section units and $K$ is the number of regressors. With computer technology of the mid 1990s, one kernel regression with $N = 9119$ and $K = 9$ took days on a 8 processor workstation. We experimented with different ways to reduce the effective sample size, including uniform sampling and binning, both with the expected effect of increasing the variance of the point estimates. For that exercise, it was effective to simply use a Gaussian instead of a quartic kernel which led to a tenfold reduction in computing time. Of course, the sample size of $N = 9119$ and $K = 9$ is trivial by today's standard. But the goal of the exercise is the same:- to efficiently analyze the data subject to resource constraints. For the Nielsen data, with $T = 260$ weeks, and $N$ in six digits, the need to efficiently analyze this data is no longer a luxury but a necessity.

An earlier literature known as 'data squashing' suggests to compress statistical information using parametric estimation. The idea is to build a squashed dataset that approximates a specific likelihood function either directly or indirectly. For example, data points having similar likelihood profiles can be deemed equivalent and merged into a single data point by taking their mean.[11] A drawback is that the squashed data points do not correspond to any unit in the sample and hence has no specific interpretation. The bigger issue is that with data now in terabytes and petabytes instead of megabytes, parametric modeling is not practical.

Consider a generic matrix $A = [A^{(1)}\ A^{(2)}\ \ldots A^{(d)}]$ with $n$ rows and $d$ columns, where each column $A^{(j)}$ is a vector of length $n$. The rank of $A$ is $r \leq \min[n, d]$. If $A$ can be factored as the product of two lower rank matrices $B$ and $C$ where $B$ is $n \times k$ and $C$ is $k \times d$, then $A$ can be stored and processed efficiently via $B$ and $C$, provided that $k$ (the numerical rank of $A$) is less than $r$. The two matrices $B$ and $C$ can be obtained by singular value decomposition (SVD) in $O(nd^2)$ operations. More efficient algorithms are available when $A$ is sparse. This is the case with the Netflix problem in which $A_{ij}$ is the user $i$'s ranking of movie $j$. But when $A$ is not sparse and $n$ or $d$ are large, the computation demand can be non-trivial.

Finding a small set of data points that provably approximate the original data of much larger dimensions has motivated researchers to look for *coresets*, or sketches of the original data. A

[11] The primary papers in this literature are Du Mouchel et al. (1999), Owen (1990), and Madigan et al. (1999). The first forms multivariate bins of the data, and then match low order moments within the bin by non-linear optimization. The second reweighs a random sample of $X$ to fit the moments using empirical likelihood estimation. The third uses likelihood-based clustering and then select data points that match the target distribution.

coreset is essentially a smaller data set that preserves interesting information of the larger data set.[12] Interest in this arise because data for video streams, images, gene expression micro arrays can be very large in size. Coresets are typically formed using algorithms without reference to the probabilistic structure of the data. Consider a high resolution color image represented by a three dimensional matrix containing the red, green, and blue pixels. A black and white image can be extracted and stored as a two-dimensional matrix consisting of the gray scale values of the image. Storing a two-dimensional matrix is of course much cheaper than a three dimensional one. The resolution of this gray scale image can be further compressed and still be of use for many purposes. In this case, the sketched matrix holds the gray scale values of the lower resolution image. How does this fit into what we do? Economic data can typically be organized in matrix form. Panel data have variables for units over time and possibly space and hence have three or more dimensions. But if we can rearrange the data into two-dimensional matrices, the data sketching algorithms can be used. For example, the rows may be units and the columns may be characteristics of the units. Or, the rows may index time, and the columns may index variables.

Hence given a $n \times d$ matrix $A$, we seek a matrix $R$ so that $R \cdot A$ would be a (linear) sketch if we want a matrix with fewer rows, and $A$ would be approximated by $A \cdot R$ if we want a matrix with fewer columns. Randomization turns out to play an important role in achieving this goal. The intuition is that any matrix $A$ can be written as a product of two matrices which can in turn be expressed as a sum: $A = PQ = \sum_{k=1}^{d} P^{(k)} Q_{(k)}$. Let $p_k$ be the probability that column $k$ is selected and define $X = \frac{1}{p_k} P^{(k)} Q_{(k)}$ for $k = 1, \dots, d$. Then $E(X) = \sum_{k=1}^{d} P(z = k) \frac{1}{p_k} P^{(k)} Q_{(k)} = \sum_{k=1}^{d} P^{(k)} Q_{(k)} = PQ$ for $z \in \{1, 2, \dots, d\}$. Hence randomly sampling the terms in the sum (with replacement) and proper rescaling will given an unbiased estimator of the product. But there are $\frac{d!k!}{(d-k)!}$ ways to choose $k$ out of $d$ columns. A systematic approach is called for.

A naive approach is to randomly sample the columns of the original matrix $A$. While uniform sampling (ie. $p_j = 1/d$) is easy to implement, it is not efficient if the data are not uniformly dispersed. For example, if the matrix $X$ contains a column that is orthogonal to the rest, and there are more columns than rows, removing it will change the rank of the matrix. Two types of randomization methods are available to deal with the non-uniformity at a low computation cost. The first method is random projections which removes non-uniformity from the data before sampling. The method approximates $A$ by linear combinations of the columns, and as such, is associated with a $R$ matrix that is dense. The second method is leverage based sampling. It takes non-uniformity into account by biasing the sample towards particular terms in the sum. It does so by choosing the columns with probability proportional to the squared length of the column. The corresponding $R$ matrix is sparse, consisting of indicators of columns of $A$ being retained.

In the next two subsections, I summarize the main idea behind these methods, referring readers to the excellent monographs of Vempala (2004), Mahoney (2011), Woodruff (2014), and Halko et al. (2011) for details. It should be clarified that the random subsampling methods considered here

[12] The term coreset was coined by Agarwal and Varadarajan (2004).

are aimed at efficient computation, and not to be confused with subsampling schemes developed for the purpose of inference that statisticians and econometricians have studied. The Frobenius and spectral norm play important roles in the discussion to follow. The squared Frobenius norm of a $n \times d$ matrix $A$ is $\|A\|_F = \sum_{i=1}^n \sum_{j=1}^d A_{ij}^2$. It is an average type criteria. The spectral norm $\|A\|_2 = \sup_{\|x\|_2=1} \|Ax\|_2$ is the largest eigenvalue of $A$. It is a worse-case type criterion.

It is useful to start with singular value decomposition (SVD). For a matrix $A$ of rank $r \geq k$ with $\sigma_j$ being its $j$-th largest eigenvalue, the SVD is $A = U\Sigma V^T$. Let $A^+ = V\Sigma^{-1}U^T$ be the pseudo inverse and $A^T$ be its transpose. Let $U_k$ be the $k$ columns of left singular vectors corresponding to the $k$ largest eigenvalues of $A$. The best low rank approximation of $A$ is

$$A_k = U_k \Sigma_k V_k^T = U_k U_k^+ A = P_{U_k} A$$

where $\Sigma_k V_k^T = U_k^+ A$ is $k \times d$. The rows of $A_k$ are the projections of the rows of $A$ onto the subspace $V_k$. For given $k$, $A_k$ is optimal in the sense that $\|A - A_k\|_F \leq \|A - D\|_F$ for any rank $k$ matrix $D$. Since $U_k U_k^+ = P_{U_k}$ is the matrix that projects on $U_k$, the residual $\|A - P_{U_k} A\|_\xi$ is minimized over all $k$ dimensional subspace for $\xi = F, 2$. The SVD has low rank approximation error of $\|A - A_k\| = \sigma_{k+1}$.

### 3.1 Random Projections

A random projection consists of taking $n$ points in $\mathbb{R}^d$ and embed (project, or map) them to a set of $n$ points in $\mathbb{R}^k$ where $k << d$. Such a projection is not useful unless it preserves the structure of the original data points. Fortunately, the influential JL Lemma (Johnson and Lindenstauss (1994)) establishes that a set of points $(u_1, \ldots, u_n)$ in $\mathbb{R}^d$ can be projected down to $(v_1, \ldots, v_n)$ in $\mathbb{R}^k$ such that for any $\epsilon \in (0, 1/2)$, and $k \geq k_0 = O(\log n/\epsilon^2)$,

$$(1 - \epsilon)\|u_i - u_j\|^2 \leq \|v_i - v_j\|^2 \leq (1 + \epsilon)\|u_i - u_j\|^2.$$

That is, random projections generate small distortions in terms of pairwise difference or the euclidean distance between points. The lemma implies that high dimensional computational problems can be solved more efficiently by first translating them into a lower dimensional space with $k$ columns, noting that $k$ depends on $n$ but not on $d$.

A sketch of the proof is as follows. Consider one vector $u$ in $\mathbb{R}^d$ and let $v = \frac{1}{\sqrt{k}} R^T u$ where $R$ is a $d \times k$ symmetric orthonormal matrix. Now $\|v\|^2 = \sum_{j=1}^k v(j)^2$ and $\|u\|^2 = \sum_{j=1}^d u(j)^2$. Hence

$$E(\|v\|^2) \;=\; \sum_{i=1}^k \frac{1}{k} E\Big[\Big(\sum_{j=1}^d R(i,j)u(j)\Big)^2\Big] = \sum_{i=1}^k \frac{1}{k} \sum_{j=1}^d E\Big[u(j)^2 R(i,j)^2\Big]] = \frac{1}{k}\sum_{j=1}^d u(j)^2 = \|u\|^2.$$

That is, the euclidean distance of the original subspace is centered around the expected value of the euclidean distance of the random projection. To bound the probability of the embedding, define

$x_i = \frac{1}{\|u\|} R_{(i)}^T \cdot u$ so that $y = \sum_{i=1}^k x_i^2 = \frac{k\|v\|^2}{\|u\|^2} = \sum_{j=}^k \frac{(R_{(j)}^T u)^2}{\|u\|^2}$. If $R$ is Gaussian, then $x_i^2 \sim \chi_1^2$. Using the properties of $\chi^2$ random variables:[13]

$$P(\Bigg( \|v\|^2 \geq (1+\epsilon)\|u\|^2 \Bigg) \quad = \quad P\Bigg( y \geq (1+\epsilon)k \Bigg) = P\Bigg( \chi_k^2 > (1+\epsilon)k \Bigg) \leq \exp\Bigg( -\frac{k}{4}(\epsilon^2 - \epsilon^3) \Bigg).$$

A similar argument shows that $P(\|v\|^2 \leq (1-\epsilon)\|u\|^2)^2 < \exp(\frac{-k}{4}(\epsilon^2 - \epsilon^3))$. Combining the results, we have, for $\epsilon \leq 1/2$,

$$P\Bigg( \|v\|^2 \notin \Bigg[ (1-\epsilon)\|u\|^2, (1+\epsilon)\|u\|^2 \Bigg] \Bigg) \leq 2\exp^{-(\epsilon^2-\epsilon^3)k/4} \leq 2\exp^{-\epsilon^2 k/8}.$$

This probability holds for $n^2$ distances between two points. By the union bound, the probability that $f$ is a $(1+\epsilon)$ embedding is at least $1 - 2n^2 \exp^{-\epsilon^2 k/8}$, which is positive for $k = O(\log n/\epsilon^2)$. The map can be found quickly, ie. in polynomial time. A projection that satisfies the lemma is known as the Johnson-Lindenstrauss (JL) transform.

An appeal of the JL transform is that it is a simple linear map and it is *data oblivious*, meaning that it can be chosen randomly with high probability irrespective of the data in the input matrix. Early work uses dense $R$ matrices. For example, a Gaussian matrix with $R(i,j) \sim N(0,1)$ is valid.[14] Subsequent work shows that the simpler matrix

$$R(i,j) = \{1, 0, -1\} \qquad \text{with prob} \qquad \begin{pmatrix} \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{pmatrix} \tag{1}$$

will still be a JL embedding with probability $1 - n^{-\beta}$, for $\beta > 0$, see Achiloptas (2003). The sampling probability of $q = 1/6$ above can changed to $q = O((\log n)^2/d)$ to further reduce computations. A good rank-k approximation of $A$ can also be obtained by choosing more than $k$ vectors.

There are many implementations of the JL- transform. See Venkatasubramanian and Wang (2011) for a review. A popular one is the so-called fast JL transform (FJLT) due to Ailon and Chazelle (2006) and Sarlos (2006). Let the sketched matrix be $B_k = AR$, where $R = DHS$

- $S$ is a $d \times k$ matrix that samples the columns uniformly at random without replacement.
- $D$ is a $d \times d$ diagonal matrix in which $D_{ii} = \{+1, -1\}$ with equal probability of 1/2.
- $H = \frac{1}{d} H_d$ is a $n \times n$ Hadamard matrix where $H_d = \begin{pmatrix} H_{d/2} & H_{d/2} \\ H_{d/2} & -H_{d/2} \end{pmatrix}$, and $H_2 = \begin{pmatrix} +1 & +1 \\ +1 & -1 \end{pmatrix}$.

Intuitively, the Hadamard transform destroys the non-uniform structure in the data. It can be thought of as a real-valued version of the complex Fourier transform that orthogonalizes the data. The orthogonalized data are re-randomized by another sparse matrix $D$. The benchmark residual

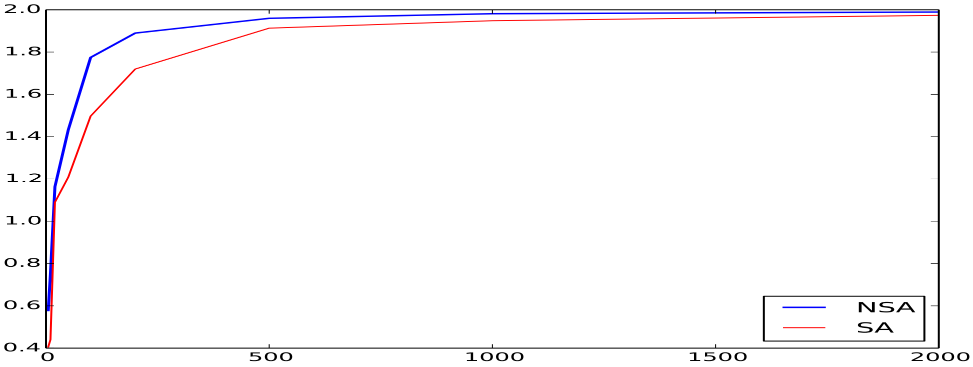
[13] In particular, $E(\exp^{\lambda x_1^2}) = \frac{1}{\sqrt{1-2\lambda}}$ for $\lambda < 1/2$.

[14] The runtime is $\Omega(kd)$ per vector, which is an asymptotic lower bound.

error is usually the best low rank approximation of $A$. It has been shown that the residual error is such that $\|A - P_{B_k}A\|_F \leq (1+\epsilon)\|A - P_{U_k}A\|_F$ with high probability. Repeating the procedure many times can boost this probability. Boutsidis et al. (2008) analyzes the approximation using the spectral norm.

To see how this subsampling scheme works on the Nielsen data, I apply random projections to the $q_{ti}$ data for beer using the SKLEARN module in Python. The entries in the $A$ matrix are the observations of either the linearly detrended, seasonally unadjusted data for week $t$ and store $i$ of $u_{ti}$ or $q_{ti}$. Using the default value of $\epsilon = 0.1$, a sketched matrix with $k = 4766$ columns is obtained from the original matrix with 64K columns. The total number of data points shrinks from over 16 million to 1.2 million.

Figure 3: Correlation of Principal Components Extracted from $A$ and $A_k$



For $j = 1, 2$, $R^2_j(B_k)$ is the $R^2$ from a regression of the $j$-th principal component extracted from $B_k$ on the first two principal components extracted from the full matrix $A$. The above plots $R^2(B_k) = R^2_1(B_k) + R^2_2(B_k)$.

Computer scientists have developed algorithms for approximating $A$ that run fast and have low worst case error bounds. But my goal is to extract the cyclical variations, which is a particular aspect of $A$. The default $\epsilon$ that yields $k = 4766$ is guided by conventional error analysis which may not be appropriate for my analysis. But how to evaluate if the sketched matrix is good or bad? For this, I turn to factor models as guide. Specifically, statistical factor analysis suggests that under some assumptions, the eigenvectors corresponding to the largest eigenvalues will precisely estimate the common factors. Let PCA1 and PCA2 be the first two principal components extracted from the $A$ matrix for the unit price data $u_{ti}$. These two components explain 0.185 and 0.157 of the variation in $A$. Let PCA1$_k$ and PCA2$_k$ be the first two principal components extracted from the $B_k$ matrix for unit price. With $k = 4766$ columns, PCA1$_k$ and PCA2$_k$ explain 0.181 and 0.163 of the variations in the $B_k$ matrix. These numbers are very similar to the ones found for $A$, which is encouraging. To obtain a more objective measure of how the common factors estimated from

$A$ compare to those estimated from $B_k$, I regress PCA1$_k$ on PCA1 and PCA2. The $R^2$ of this regression is denoted $R_1^2(B_k)$. Similarly, PCA2$_k$ of $B_k$ is regressed on PCA1 and PCA2 to give $R_2^2(B_k)$. Let

$$R^2(B_k) = R_1^2(B_k) + R_2^2(B_k).$$

This quantity has a maximum of two since $R_1^2(B_k)$ and $R_2^2(B_k)$ each has a maximum value of one. I interpret $R^2(B_K)$ as a summary statistic of how close is the space spanned by the first two principal components of the full matrix and the first two components of the sketched matrix.

To explore the sensitivity of the estimated common factors to $k$, I compute PCA1$_k$ and PCA2$_k$ from $B_k$ for different values of $k$. The line labeled NSA in Figure 3 shows that $R^2(B_k)$ is above 1.9 when the dimension is over 1000. The $R^2$ labeled SA indicates that more columns are needed to sketch the seasonally adjusted data (that will be constructed in the next section). The exercise is also repeated by comparing the span of three instead of two principal components. As expected, the more components we are interested in, the bigger must $k$ be for $R^2(B_k)$ to be close its maximum achievable value. For the beer data, $k = 1000$ is enough to give an $R^2(B_k)$ of over 2.7 when the maximum is three. The $R^2(B_k)$ criterion only compares the top (instead of all) principal components of $A$ with $B_k$, which may not be optimal on any ground. But it seems that while the existing generic algorithms do a good job preserving the features of the largest eigenvectors, they can be further improved to suit specific objectives.

## 3.2 Leverage Score Sampling

Random projections produce sketches of a matrix by removing the non-uniformity in the data. The columns of the sketched matrices are linear combinations of columns of the original matrix and hence lack interpretation. It is sometimes useful to sketch a matrix by selecting specific columns rather than forming linear combinations. For example, an eigen-gene has no meaning in gene array analysis, nor is a linear combination of barcodes meaningful.

The problem of efficiently and accurately finding a matrix consisting of exactly $k \leq r$ columns of $A$ is known as the *column subset selection* problem (hereafter CSSP). Let $C_k = A \cdot R$ be the sketched matrix. In a CSSP, $R$ is $d \times k$ sparse matrix consisting of indicators of the columns being selected. There are two approaches to construct $C_k$. The linear algebra community proceeds by noting that the problem reduces to selecting columns in the upper triangular matrix of the $QR$ decomposition of $A$. Solutions can be obtained using the rank revealing methods (RRQR) first developed in Golub (1965).[15] Methods within this class differ in how the columns are pivoted, but they are fundamentally deterministic in nature, see Gu and Eisenstat (1996).

In contrast, computer scientists take a random approach. Frieze et al. (2004) suggests to sample the columns of $A$ with replacement using probabilities $p_j$ that depend on the Euclidean norm of

[15] A rank-revealing factorization finds the numerical rank of a matrix, or the index $r$ such that $\sigma_r >> \sigma_{r+1} = O(\epsilon)$, $\epsilon$ is machine precision. If $\Pi$ is a column permutation matrix, $A\Pi = Q\begin{pmatrix} R \\ 0 \end{pmatrix}$ is a rank revealing $QR$ factorization.

the column of $A$, ie. $p_j = \frac{\|A^{(j)}\|_2^2}{\|A\|_F^2}$. Once the columns are picked, the sketched matrix is obtained by projecting onto the subspace spanned by $k$ columns of $A$. While the run time is fast, the additive error rate is not satisfactory and the probabilities are not invariant to normalization. Furthermore, the original matrix may not be selected when $k = d$.

An improved sampling scheme known as CUR, proposed in Drineas et al. (2008), is to replace Eculidean norm by a direct measure of where information in $A$ is centered. The idea is to keep column $j$ with probability $\min(1, c \cdot p_j)$ for some $c = O(k \log k\epsilon^2)$, where

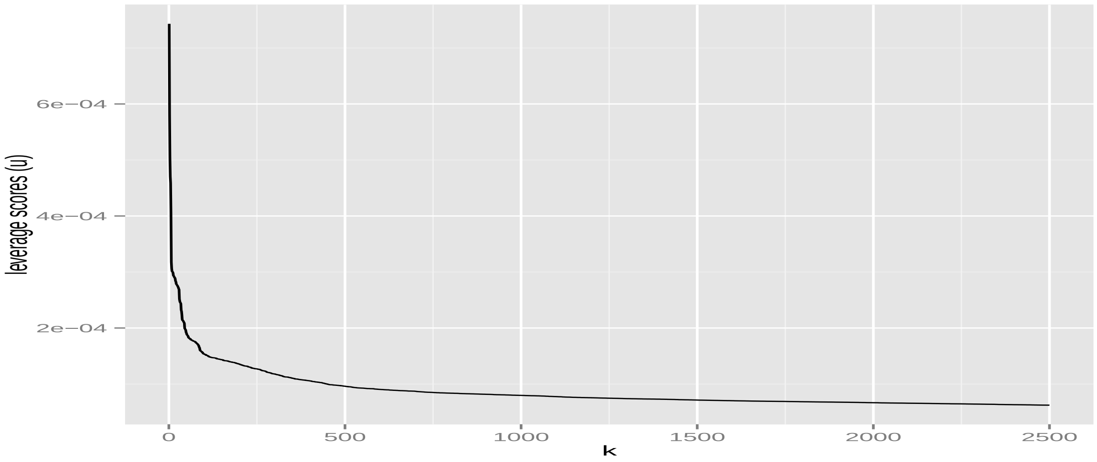
$$p_j = \frac{1}{k}\|(V_k^T)^{(j)}\|_2^2. \tag{2}$$

The normalization by $k$ ensures that $p_j$ sums to one. Boutsidis et al. (2009) suggests a two step CSSP algorithm that further improves upon the CUR algorithm. In the first step, a randomized algorithm is used to oversample $k_1 = O(k \log k) > k$ columns, where column $j$ is selected with probability $\min(1, c \cdot p_j)$. In step two, a deterministic RRQR algorithm is used to pick exactly $k$ columns from the rescaled $n \times k_1$ matrix to form $C_k$. The time complexity of the algorithm is $O(nd^2)$ and an error bound of $\|A - P_{C_k}A\|_F \le O(k\sqrt{\log k})\|A - P_{U_k}A\|_F$ can be achieved with high probability. An advantage of CUR and CSSP is that the columns of the sketched matrix are those of the original data. Hence unlike the method of random projections, the representation of the data is preserved.

But what exactly is $p_j$ and why is the resulting error rate smaller than the one obtained when $p_j$ is defined from the Euclidean norm of $A$? Intuitively, we know from singular value decomposition that the Euclidean norm of $A$ is a convolution of $U_k$, $V_k^T$, and $\Sigma_k$. The subspace information $V_k$ are more precise indicators of where information in $A$ is concentrated. Hence when used to define $p_j$, they select columns that contain more relevant information about $A$.[16] It turns out that $p_j$ defined in (2) can be motivated from a regression perspective. For the linear model $y = X\beta + e$ where $X$ is full column rank, the projection (hat) matrix is $H = X(X^TX)^{-1}X^T$, and the fit is $\widehat{y} = Hy$. As is well known, $i$-th the diagonal element of $H = UU^T$, say, $H_{ii}$, measures the influence, or leverage, of observation $i$. Points with a high leverage have more influence on the fit, hence $H_{ii}$ can be used to detect which observations (or rows) are outliers. Here, we are interested in column selection. Hence the leverage scores are defined by the right eigenvectors $V^T$. By using the leverage scores to determine $p_j$, leverage score sampling favors columns that exert more influence on $A$. Accordingly, $p_j$ defined in (2) is known as (normalized) statistical leverage scores. These probabilities define an importance sampling distribution. It was first used in Jolliffe (1972). By using the left instead of the right singular vectors (ie. $U$ instad of $V$) to define the sampling probabilities, leverage scores can also be used to select rows. Selecting rows and columns simultaneously is, however, a harder problem since a set of good columns and a set of rows selected separately to have some desired characteristics may no longer have the desired features when put together to form a new matrix.

[16] Note that the $V^k$ required to compute $p_j$ can be obtained without a full svd.

Figure 4: Leverage Scores: Unit Price, Beer

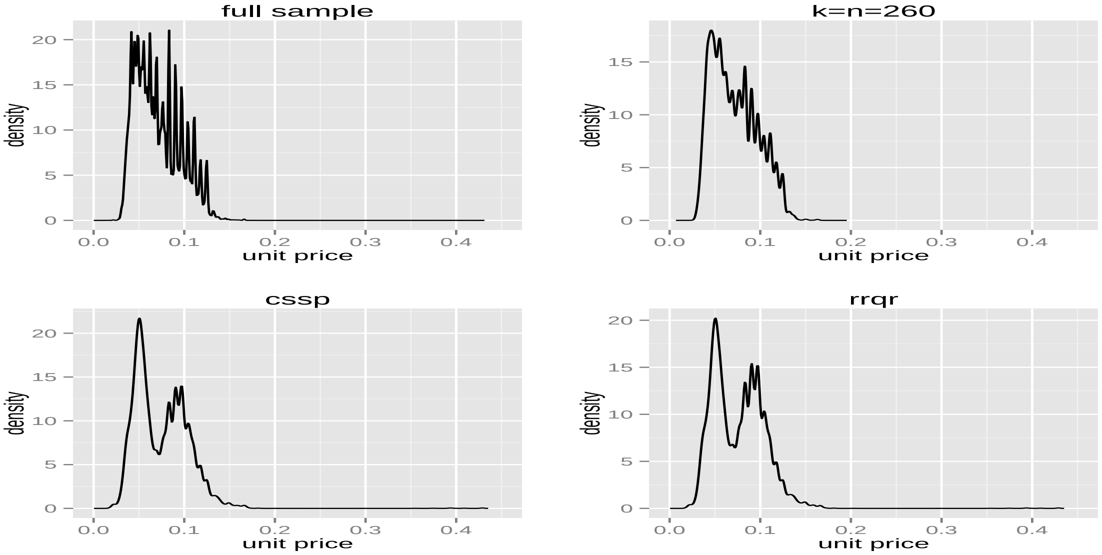


I use the unit price data to evaluate the column selection procedures because the quantity data have strong seasonal variations which could affect the leverage scores. Figure 4 plots the largest 1000 leverage scores of the unit price data. It shows that only about 200 out of 65K units have large leverage scores, suggesting a non-uniform structure in the data. Because the data have a non-uniform structure, random sampling should not perform well. To see how the sampling scheme affects features retained in the sketched matrix, the top left panel of Figure 5 presents the full sample histogram of unit price. The histogram has a long right tail. As seen from Figure 2, there are large spikes in the individual unit price data, probably reflecting discount sales. Next, I randomly select $k = n$ (where $n$ is 260 in this example) columns. The corresponding density (denoted pct-0) has a short right tail and almost looks symmetric, quite unlike the full sample density. I also randomly select a fraction $\frac{x}{1000}$ of the sample and label it 'pct-x'. For example, pct-10 indicates one percent of the sample is retained. These densities are similar to the one shown for pct-0 unless $x$ is very big. The purely random CUR algorithm has a bimodal density but still has a short right tail (not shown to conserve space). The CSSP gives a density that has a long right tail, much like the feature of the original matrix. It is also smoother, suggesting that the random sampling also reduces local variations. The RRQR algorithm does not have a leverage interpretation but preserves the shape of the density of the original data quite well. Hence, when the data are not uniformly dispersed, how the coreset is formed can affect what features in the original data are preserved. Guided by Figure 3, I also extract two principal components from the matrices sketched by CSSP with $k = 4(n-1) = 1036$ columns.[17] One can hardly distinguish

[17] Having $k = 4n$ columns is not an issue for random projections. This is a problem for CSSP because if $A$ has rank $r = \min(n, d)$, $C$ necessarily has rank $k \leq r$. For this data, $\min(n, d) = n = 260$, so the procedure will select no more than $n$ columns, discarding more information than we can afford to use. I explored several to remedy this problem. The first is uniformly sampling additional $3n$ columns, and concatenate them to the columns selected by

between the principal components constructed from the full and the sketched matrix. These are not shown.

Figure 5: Density of Unit Price, Beer



This section has explored the possibility of using coresets instead of the full sample to summarize features of the data. Random sampling can speed up big data analysis but the sampling scheme can affect what characteristics of the original data are preserved, and hence what can be uncovered from the subsamples. The two approaches considered – random projections and leverage score – account for the non-uniform structure of the data in a non-probabilistic way. In spite of the lack of an explicit model, my crude analysis suggests that the two approaches can provide sketches that preserve features of the leading principal components reasonably well.

The results from this investigation are encouraging, but a rigorous analysis is needed to evaluate these algorithms in terms of economic objects of interest, such as trends and cycles, index numbers, and consumption distributions. Here, I have used large dimensional factor models for guidance, even though my data may not be stationary, and the factor structure may not be strong. Furthermore, there may be unique features in the economic data that we want to preserve. For example, we may want to incorporate prior information to ensure that certain UPCs are included, or we may want to oversample stores in certain location. In such cases, leverage scores will no longer be the only basis for columns selection. Often, we have data on multiple variables that are not independent, such as price and quantity, and it would seem reasonable to create subsamples of the two variables jointly instead of one data matrix at a time. It seems unlikely that generic algorithms will deliver the best results for a specific objective. Input of economists will likely be needed to address issues specific

leverage based sampling. The second is to stack up four matrices each of dimension $n \times d$ so that we can select $4n$ columns. Our four matrices are unit price, quantity, transactions price, and value. Either way, the resulting principal components have properties comparable to those based on random projections.

to economic data. The challenge is how to combine aspects of the algorithmic approach, stochastic modeling, and economic information to make big data analysis feasible and more efficient.

Efficiently forming coresets is an active area of research by data scientists. Algorithms for $L_p$ regressions with good run time and desirable worse-case error bounds are already available; see, for example, Drineas et al. (2011), Maillard and Munos (2012). This can be tremendously helpful in structural analysis using big data. However, computational efficiency may not lead to statistical efficiency. Evidently, statisticians are also taking an interest in understanding these estimates in terms of their bias and mean-squared error. Work by Ma et al. (2014) and Li et al. (2006) is perhaps a sign of a merge of computational and statistical thinking.

## 4 Seasonal Adjustment

Many products exhibit seasonal sales, as illustrated in Figures 1 and 2 for the quantity of beer sold. An economic time series $z_{ti}$ can be expressed as the sum of a trend ($d_{ti}$), a cycle ($c_{ti}$), a seasonal ($s_{ti}$), a holiday ($h_{ti}$), and an irregular component ($e_{ti}$):

$$z_{it} = d_{ti} + c_{ti} + s_{ti} + h_{ti} + e_{it}$$

Macroeconomists typically focus on the business cycle component $c_{ti}$. But as all components on the right hand side are latent, there is always the possibility that seasonal and holiday adjustments can distort $c_{ti}$. Wright (2013) suggests that because the Great Recession had the sharpest downturn occurring between November 2008 and March 2009, the seasonal filter might have treated the downturn as a bad winter, resulting in a difference of 100,000 jobs reported for monthly non-farm payroll. Indeed, the business cycle components in the Nielsen scanner data will necessarily depend on how we handle $s_{ti}$ and $h_{ti}$.

Nowadays, monthly and quarterly data for the U.S. and Canada are most likely adjusted by the X-12 or X-13 programs maintained by the Census Bureau, and implemented in popular software packages. These filters remove periodic variations using a constant parameter ARIMA model one series at a time. It also adjusts for outliers, easter, labor day, and thanksgiving effects. In Europe, it is more common to use the TRAMO/SEATS program[18] that can estimate the different components of the time series simultaneously.

The seasonal adjustment of weekly data is more complicated than the adjustment of monthly or quarterly data. At issue is that the Gregorian (solar) calendar has a 400 year cycle (or 20,871 weeks). In this cycle, 329 years have 52 weeks, 303 years have 365 days, 71 years have 53 weeks, and 97 (leap) years have 366 days. As a consequence, the major holidays such as Christmas, Easter,

[18] The X-12/X13 filters are based on the X-11 protocol developed at Statistics Canada in the mid 1970s. SEATS is the Signal Extraction in ARIMA Time Series procedure. An effort is underway to standardize the seasonal adjustment process. See ESS Guidelines on Seasonal Adjustment, Eurostat Methodologies and Working paper. The Census Bureau now provides a X-13ARMIA-SEATS that uses a version of the SEATS procedure developed at the Bank of Spain,.

Labor Day, and Thanksgiving do not fall on the same day every year.[19] The timing of events such as Superbowl, which generate economic activities for reasons unrelated to the business cycle, also changes from year to year. As seen from Figure 2, values of the aggregate data $\overline{q}_t$ are typically higher during the summer weeks, but the spikes do not occur on exactly the same week each year. Furthermore, even though seasonal effects are present at both the aggregate and the unit (store-UPC) level, they do not necessarily spike in the same week. Variations that are not exactly periodic cannot be removed simply by differencing.

Several approaches have been suggested to seasonally adjust weekly data. One is the CATS-D regression approach proposed in Pierce et al. (1984) to remove deterministic seasonal variations. The program allows for several U.S. holidays and additional ones can be specified by the user. In 2002, the Bureau of Labor Statistics replaced CATS-D by the CATS-M program of Cleveland and Scott (2007). The CATS-M uses a locally weighted regression to allow the seasonal factors to change over time. Also available is the structural state space approach of Harvey and Koopman (1993) and Harvey et al. (1997). The parameters of the model need to be tuned to the series in question. Chevalier et al. (2003) removes the holiday and seasonal effects in the weekly data of a large supermarket chain in Chicago. In general, nonparametric and non-linear regression analysis are difficult to implement when there is substantial product and spatial heterogengeity. Parameters tuned to achieve the desired effect for a particular series may not work well for all series. I need a practical, fairly automated approach that can remove 'enough' seasonal variations so as to extract the cyclical component in the data. The next subsection considers a bottom-up approach that adjusts the data at the unit level.

### 4.1 Adjusting the Individual Series

Weekly data are typically recorded on a particular day of the week, and this is also true of the Nielsen scanner data. Let observation $t$ be defined by a triplet (week, month, year). If the sample starts in 2006-01-07, then $t = 59$ corresponds to 2007-02-17, which is week three in month two of year 2007. For the sake of discussion, let $q_{ti} = q_{w_\tau,i}$ be the log of quantity sold by unit $i$ in period $t$, which is week $w$ of year $\tau$. Also let $\overline{q}_t = \overline{q}_{w_\tau}$ be the log of total quantity sold in the same week. Motivated by the CATS-D approach of Pierce et al. (1984), I specify the seasonal component as

$$\begin{aligned} s_{ti} &= \sum_{v=1}^{k_y} \Big[ a_{iyv} \sin(2\pi v \cdot y_t) + b_{iyv} \cos(2\pi \cdot v y_\tau) \Big] \\ &\quad + \sum_{v=1}^{k_m} \Big[ a_{mi} \sin(2\pi v \cdot m_\tau) + b_{mv} \cos(2\pi \cdot v m_\tau) \Big] + \vartheta_1 \text{TEMPMAX}_{ti} + \vartheta 2 \cdot t, \end{aligned} \tag{3}$$

[19] In the U.S., the major holidays are Christmas, New year, Easter, Labor day, Memorial day, April 15 tax day, July 4, Presidents day, Thanksgiving, MLK day, Veterans day, and Columbus day.

where

$$y_t = \frac{\text{day of year}_t}{\text{days in year}_t}, \qquad m_t = \frac{\text{day of month}_t}{\text{days in month}_t}.$$

As in Pierce et al. (1984), the sine and cosine terms pick up the purely deterministic seasonal variations. The time trend linearly detrends the data. It remains to find a simple way to control for stochastic seasonal variations without having to estimate ARMA models and to handle to holiday effects in an automated way.

I capture the stochastic seasonal variations using the climate data collected by the National Ocean and Atmospheric Administration (NOAA). Precisely, the NOAA data contains the latitude and longitudinal coordinates of climate stations. From this information, the county in which each station is located can be identified. The climate data are then merged with the Nielsen data using the county code of the store.[20] After experimenting with maximum and minimum temperature, snow, and precipitation, only maximum temperature is used. Hence, the variable $\text{TEMPMAX}_{ti}$ in (3). This overcomes the problem that parts of the country have no snow, and that the maximum and minimum temperatures are fairly collinear.

Removing the holiday effects requires finding the weeks with unusually high transactions, one year at a time, and then positioning dummy variables to remove them. The challenge is that holiday effects can differ between products. I treat holiday effects as 'common features' of the data, the reasoning being that national holidays occur on the same day irrespective of location. Thus, I devise an algorithm that exploits the rich cross-section information within a product type to let the data determine the dates with unusually high volume of transactions. Let

$$h_{ti} \;=\; \sum_{\tau=2006}^{2010} c_\tau \cdot 1(\mathbb{T}_\tau(t) \in \mathbb{A}) + \sum_{\tau=2006}^{2010} \sum_{v=1}^{n_H} \delta_{i\tau\, v} 1\Big(\mathbb{T}_\tau(t) = \mathbb{H}_\tau(v), \mathbb{T}_\tau(t) \notin \mathbb{A}\Big) \tag{4}$$

where $\mathbb{T}_\tau(t)$ is a function that returns the week in year $\tau$ associated with $t$, $\mathbb{H}_\tau$ and $\mathbb{A}$ are two sets of dates based on individual and aggregate quantities sold, respectively. Note that $\mathbb{H}_\tau$ is year specific while $\mathbb{A}$ is not. The construction of these variables is now explained in further detail.

Turning first to $\mathbb{H}_\tau$, we have, for each $i$, 260 weekly observations $q_{w_\tau i}$. Let $w^*_{\tau i} = \max_{w_\tau} q_{w_\tau i}$ be the week in which the number of units sold by store $i$ in year $\tau$ was highest. Let $\mathbb{H}_\tau$ be a $n_H \times 1$ vector consisting of the top $n_H$ weeks in year $\tau$ as indicated by the cross-section distribution of $w^*_{\tau i}$. Next, I rank weekly aggregate sales $\overline{q}_{w_\tau}$ for year $\tau$, giving a score of one to the week with highest total units sold in year $\tau$, two to the week with the second highest total units sold, and so on. Since we have five years of data, the best possible total score any given week can accumulate is five. The $n_A$ weeks with the highest total score are collected into a vector $\mathbb{A}$. The dates in $\mathbb{A}$ need not be the same as those in $\mathbb{H}_\tau$, but they may overlap. To avoid multicollinearity, I only use those

[20] Source: `http://www1.ncdc.noaa.gov/pub/data/ghcn/daily/readme.txt`. See Menne et al. (2012) `http://cdiac.ornl.gov/epubs/ndp/ushcn/daily_doc.html` for details of the data. For counties with more than one station, the climate data are averaged. Boldin and Wright (2015) Also used climate data recorded at the 50 largest airports to construct proxies for unseasonal weather.

dates in $\mathbb{H}_\tau$ not already in $\mathbb{A}$.

To illustrate, consider the beer data. The algorithm returns $\mathbb{A} = (27, 22, 21, 26, 25, 36, \ldots)$ which indicates that at the aggregate level, more units of beer are sold around July 4th (week 27) and memorial day (week 22) than any other week in the year. The first six entries of $\mathbb{H}_\tau$ are

$\mathbb{H}_\tau$

| | Units of Beer Sold: Week | | | | | |
|---|---|---|---|---|---|---|
| Year $\tau$ | Best | Second | Third | Fourth | Fifth | Six |
| 2006 | 27 | 51 | 22 | 47 | 18 | 26 |
| 2007 | 27 | 51 | 22 | 47 | 18 | 36 |
| 2008 | 27 | 51 | 22 | 28 | 36 | 1 |
| 2009 | 27 | 51 | 22 | 28 | 1 | 21 |
| 2010 | 27 | 22 | 26 | 18 | 47 | 6 |

At the store level, $\mathbb{H}_\tau$ indicates that the high volume weeks are around July 4th, Christmas, and Memorial day. The only surprises in these dates are perhaps the omission of dates around Superbowl and the inclusion of week 18, which happened to be the Easter weekends. But according to Nielsen, Superbowl beer sales only ranked 8th, behind Easter. Hence the dates identified by the algorithm are quite sensible. In the seasonal adjustment regression, I use $n_A = 6$ and $n_H = 3$. Since weeks 22 and 27 are in both $\mathbb{A}$ and $\mathbb{H}_\tau$, these two weeks are dropped from $\mathbb{H}_\tau$ to obtain a more parsimonious specification. It should be made clear that $\mathbb{H}_\tau$ and $\mathbb{A}$ are product specific. For example, meat and wine sales peak around thanksgiving and Christmas (weeks 45 and 51), but not around July 4th.

To recapitulate, I have augmented the CATS-D model to include data driven holiday dummies, and I use observable variations in climate to bypass ARMA modeling of the latent stochastic seasonal variations. Equation (3) can in principle be estimated using a fixed-effect regression, but there are two problems. First, I have 65K units each with 260 weeks of beer data, hence 16 million data points for $q_{ti}$ alone. The pooled regression is memory intensive because of all the seasonal variables involved. More important is that pooling constrains the seasonal effects to be homogeneous across units. This is restrictive because stores in Florida and may not have the same seasonal pattern as stores in, say, Wisconsin. The periodic spikes in the residuals of the pooled regression suggest that pooling failed to remove the seasonal and holiday effects. The variety at the spatial and product levels that make the data interesting also make preprocessing the data difficult.

Both considerations suggest to estimate $s_{ti}$ and $h_{ti}$ on a series by series basis. This also bypasses the need to account for firm level heterogeneity in a pooled regression, which would have required weighing the observations of each unit by its volume or sales. If I had used the sine and cosine functions alone to model seasonality as in CATS-D, the residual maker matrix would have been the same across units. Adding the climate data makes the regressor matrix unit specific. In spite of this, it takes less than an hour to do 65K regressions. Each regression yields a $\overline{R}^2$ which is a convenient indicator of the importance of the seasonal effects. Most of the $\overline{R}^2$s for the beer regressions are between 0.2 and 0.3, but a few are above 0.85. Now some products (such as baby

Figure 6: $u_{it}$ and $q_{it}$ (SA): Pet Food and Beer

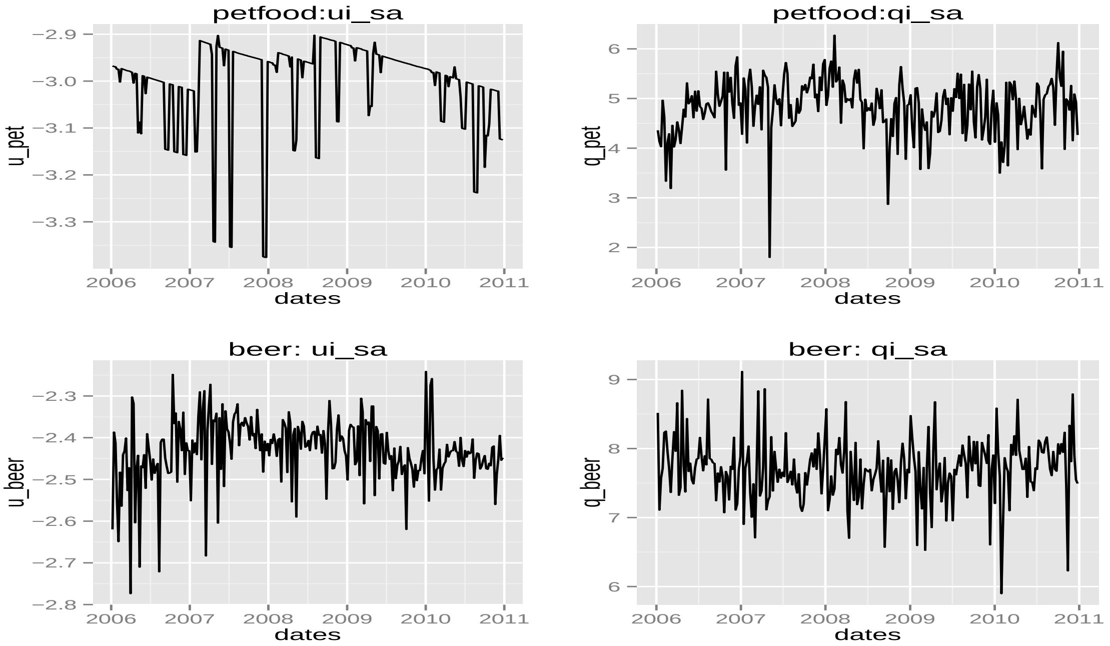
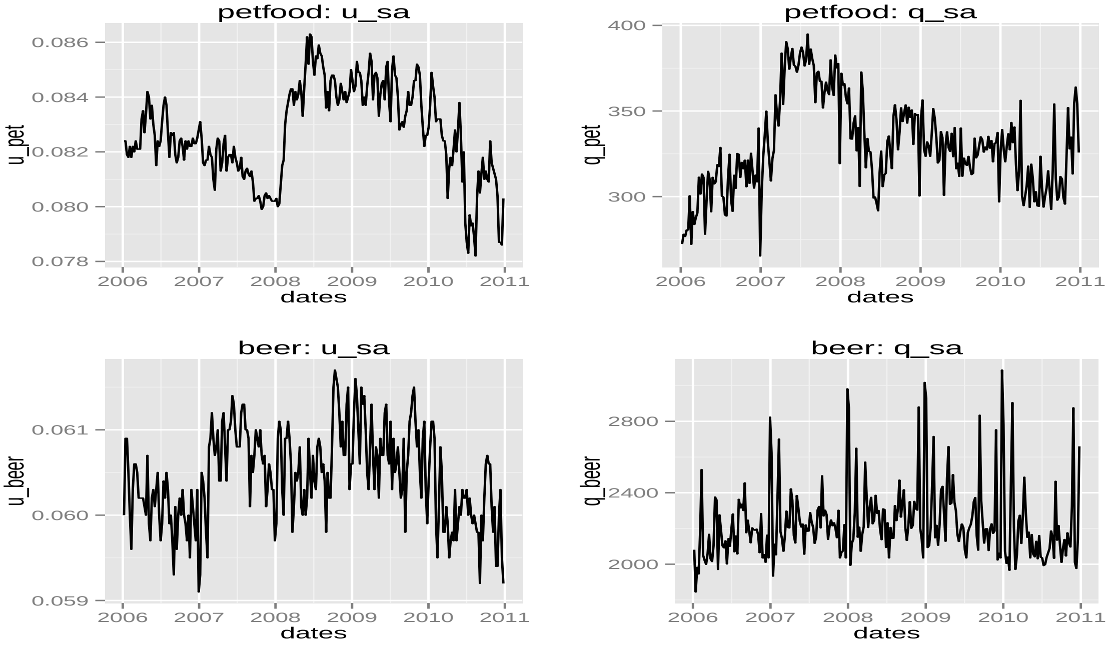


Figure 7: $\overline{u}_t$ and $\overline{q}_t$ (SA): Pet Food and Beer



food and diapers) exhibit weak seasonal effects. To avoid spurious seasonal filtering, the residuals from (3) are used as adjusted data only if the $\overline{R}^2$ of the regression exceeds a threshold, which I set to 0.1. Otherwise, the adjusted series is simply the unadjusted data. Precisely, the sum of the residuals from the seasonal adjustment regression and $\overline{q}_i$ is taken to be the log seasonally adjusted series $q_{ti}^{sa}$.[21] Exponentiating $q_{ti}^{sa}$ and sum over $i$ gives an aggregate, seasonally adjusted series, $\overline{q}_t^{sa}$ that is constructed from bottom-up.

Figure 6 shows the seasonally adjusted data for the same two units as in Figure 1. These two series are chosen because they have the highest $\overline{R}^2$ in the seasonal adjustment regression. The raw pet food data have little seasonal variations to begin with and the regression preserves the properties. For beer, the adjusted data at the unit level also show little seasonal variations. However, as seen from Figure 7, the aggregate data for beer still exhibit seasonal effects even though they are less pronounced than the raw data shown in Figure 2.

## 4.2 The Cyclical Component at the Aggregate Level

Large dimensional factor analysis suggests that if there are $r$ common factors the principal components corresponding to the $r$ largest eigenvalues of the data matrix should consistently estimate the space spanned by the factors under some assumptions. With the hope that the cyclical component will be one of the top components, I analyze the first three principal components in each of the products, fixing the sign so that all components have a trough around the Great Recession. Since the data being analyzed are in level form, the common variations can have a trend and a cyclical component. By cycle, I mean the stationary (mean-reverting) common variations that display at least a peak and a trough.

Figure 8 displays the first three principal component of six products. The first component PCA1-SA is in black, the second component PCA2-SA is in darkgreen, and the third component PCA3-SA is in blue. For all six products, the first component is always highly persistent with a trough around mid-2008. I interpret this as the common trend in the data. The interpretation of the second and third components is more tricky. The second and third eigenvectors of the eggs data appear to random noise, suggesting that there are no common variations in eggs beyond PCA1-SA. For baby food, the second component has large variations but lack interpretation, though the third component appears cyclical. For beer and meat, the second component has strong seasonal variations but the third component is cyclical. For pet food and foreign-wine, the second and third components are both cyclical. The common variations are strongest in beer. The three principal components explain about 0.18 of the variations in the data. A compact summary of the first three principal components is as follows:

[21] These residuals are persistent but mostly, stationary, which is why the regression is specified in level form. For longer samples, a first difference specification might well be needed.

| pca | 1 (black) | 2 (green) | 3 (blue) | peak-cycle | trough-cycle | common var. |
|---|---|---|---|---|---|---|
| eggs | trend | noise | noise | na | na | 0.14 |
| baby food | trend | noise | cycle | 2007-11-03 | 2009-06-13 | 0.11 |
| meat | trend | seasonal | cycle | 2007-09-15 | 2009-04-25 | 0.12 |
| beer | trend | seasonal | cycle | 2007-04-07 | 2009-06-13 | 0.18 |
| pet food | trend | cycle | cycle | 2007-03-24 | 2008-06-21 | 0.17 |
| foreignwine | trend | cycle | cycle | 2007-04-14 | 2008-10-18 | 0.14 |

The Nielsen weekly data presents a unique opportunity to study consumer behavior around the Great Recession of 2008. I explore the cyclical component of pet food, beer, and foreign-wine. These are shown in Figure 9, along with the 'Retail and Food' seasonally adjusted monthly series produced by the Census Bureau and obtained from FRED as RSAFS. The RSAFS series (in red) takes a big dive in the third quarter of 2008 and reaches its trough in the first quarter of 2009. The decline is steepest around 2008-09-15, right around the time when Lehman Brothers fell, but well after Bear Stearns collapsed on 2008-03-16. The foreign-wine series reaches its peak on 2007-04-14 and its trough on 2008-10-18. The pet food series is at its peak in 2007-03-24 and reaches bottom around 2008-06-21, similar to the peaks and troughs found for beer, but about one year ahead of the peak and trough exhibited in the RSAFS series.

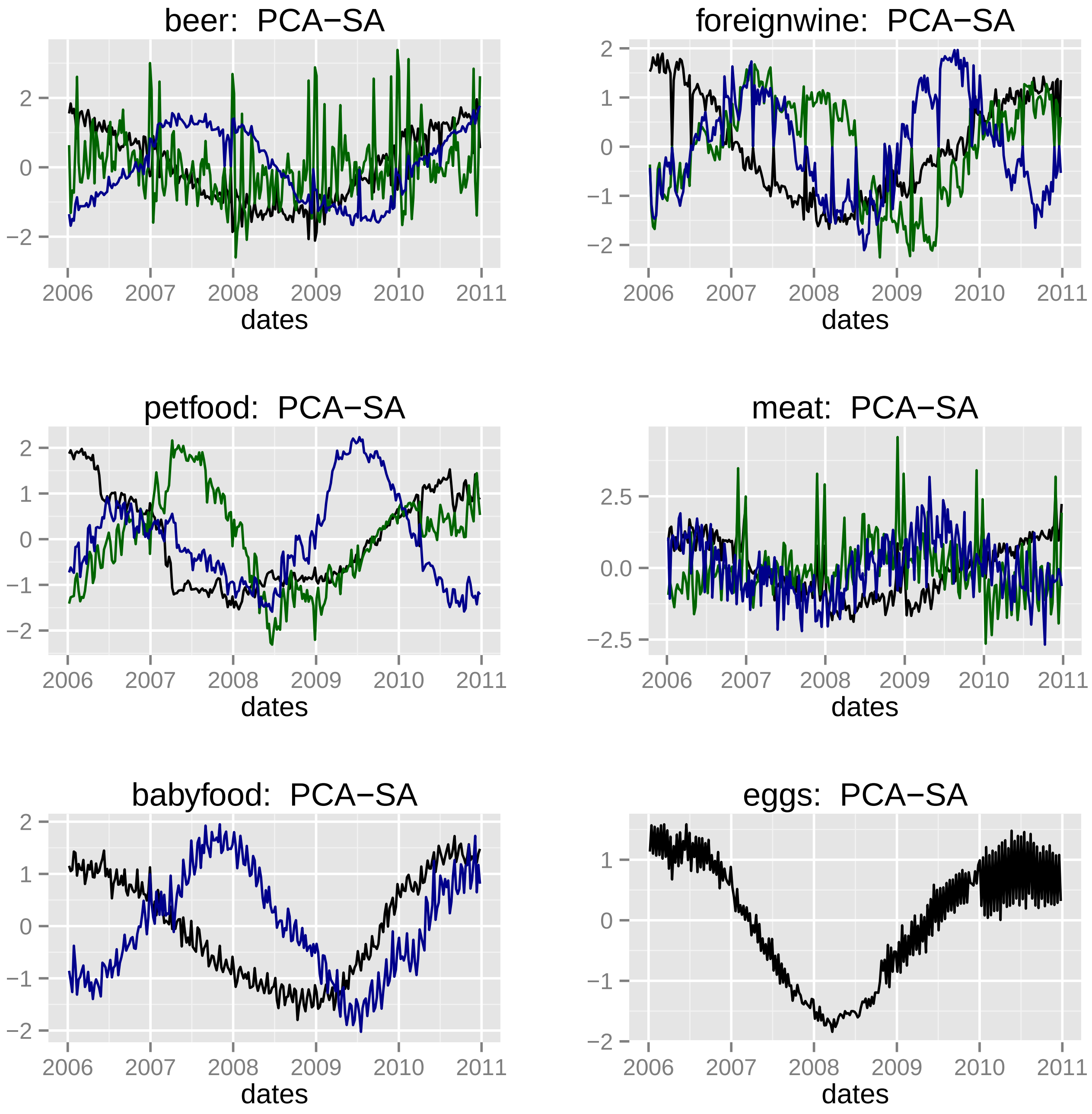
To see if there is an agreement between consumer confidence and action, I compare the cyclical component with the Rasmussen Consumer Index.[22] This index is the seven day moving-average obtained from polling users about their expectations and confidence. I then select those polling days that the Nielsen data are also available, and for the few days with missing data due to holidays, I use the data polled on the closest day available. Figure 10 shows that the cyclical component of the three series track the Rasmussen index (in brown) quite well. Spending on these products is high when confidence is high, and low when confidence is low. At least for these products, actions and sentiment seem to be in sync.

In summary, I find strong co-movements across units in the seasonally adjusted data of just about every product being analyzed in the form of a highly persistent principal component, and which I interpret as a common trend. I also find a common cyclical component that precedes the big downturn of 2008. That some of these cyclical components tend to lead the aggregate retail sales data could be of interest in the monitoring of economic activity. However, the fact that the aggregate data still exhibit seasonal variations is disappointing. Removing the seasonal variations at the individual level apparently did not lead to an aggregate series that is rid of seasonal variations. This raises the question of whether I should have adjusted the aggregate data directly, or in other words, top-down instead of filtering the seasonal effects from bottom up.

One thought is to treat the seasonal variations as a common factor. I can then directly look for the trend, the cycle, and the seasonal factors from the seasonally unadjusted data. This is

[22] I thank the Rasmussen Group for providing me with this data.

Figure 8: PCA-SA



easy to implement as I just let the method of principal components do its work. To explore this methodology, I re-analyze the seasonally unadjusted data of pet-food, beer, and foreign-wine. Figure 11 shows the three principal components in black, darkgreen, and blue, respectively. Three results are noteworthy. First, the PCA2-SA and PCA2-NSA series for pet food are similar. This is reassuring since the pet food data have small seasonal variations; my seasonal adjustment has preserved the variations in the raw data. Second, the PCA1-NSA series (in black) for beer and foreign-wine indeed have strong periodic movements. Observe that PCA2-NSA (green) and PCA3-NSA (blue) for beer shown in Figure 11 resemble PCA1-SA (black) and PCA2-SA (green) shown in Figure 8. The principal components of the adjusted data are shifted up compared to the the unadjusted data because there is no longer the need to accommodate the seasonal factor. This is good news because whether I use a model to seasonally adjust the data or let the method of principal components find these seasonal variations, the trend and the cycle in the raw data are similar. The third observation is that seasonal effects show up in more than one principal component. Notably, PCA3-NSA for foreign-wine (in blue) still has periodic spikes.

This top-down approach shows promise but needs to be further developed. The principal components being identified evidently depend on the relative importance of the trend, cycle, and seasonal variations, and these are are product specific. To successfully isolate the common variations from the seasonally unadjusted data directly, I would need a way to systematically associate the principal components with the trend, the cycle, and the seasonal variations on a product-by-product basis. This is not so straightforward when the different variations may not be mutually uncorrelated.

## 5 Concluding Comments

This paper has set out to better understand what makes big data analysis different. I used four terabytes of Nielsen scanner data as case study, with the aim of analyzing the business cycle variations around the Great Recession of 2008. The task is non-conventional mainly because the memory constraint limits how much information can be processed at a time, the data are highly heterogeneous, and that weekly seasonal variations need to be removed. There was a bit of trial-and-error in the exercise, but most of it is learning-by-doing.

Data scientists have developed many tools to accommodate the 3V characteristics of big data. This paper has focused on the ones used in data-preprocessing and found subsampling algorithms to be flexible with the potential to be very useful in economic analysis. Most of these algorithms are, however, not developed with economic data in mind. It will take some effort to properly integrate them into our analysis. There is definitely a need for new methods that are computationally efficient and statistical optimal. Bridge the gap between the econometric and the algorithmic approaches to data modeling will likely be a multi-disciplinary endeavor.

When the database is so massive, it is inevitable that some information will have to be dropped. In the end, I only analyzed a tiny fraction of the data available. Nonetheless, for the purpose of

Figure 9: RSAFS vs. Pet-Food, Foreign Wine, Beer

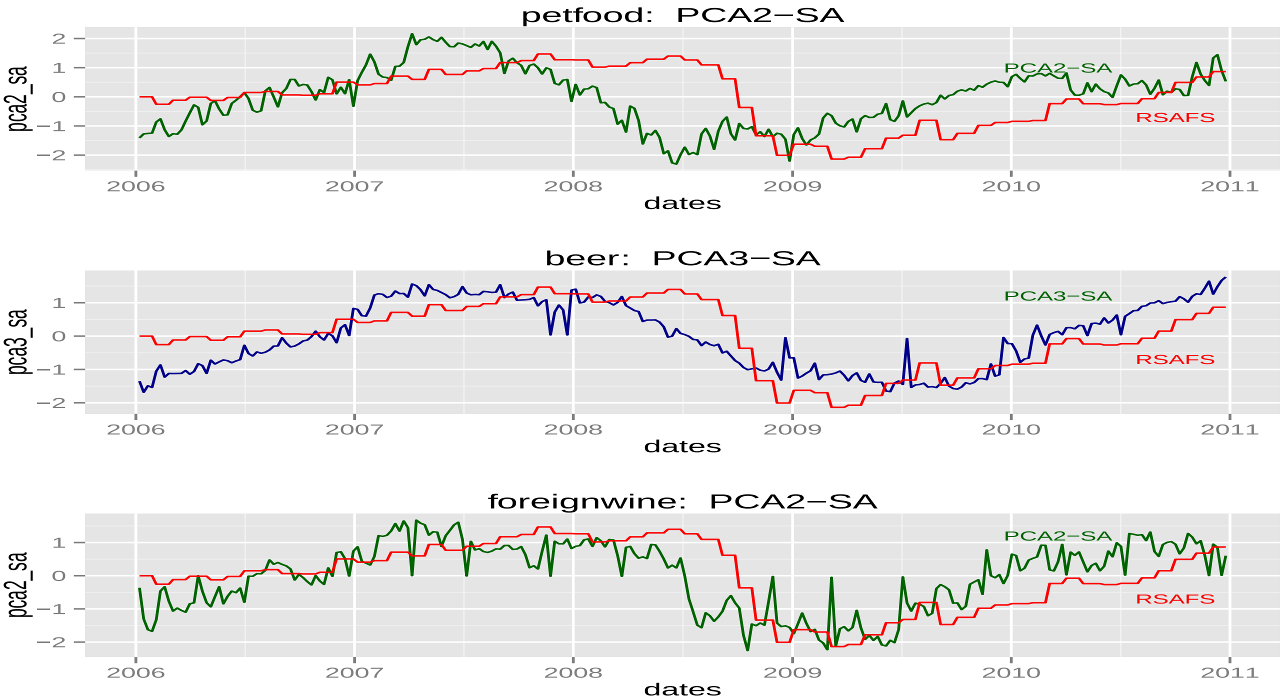


Figure 10: Confidence vs. Pet-food, Beer. Foreign Wine

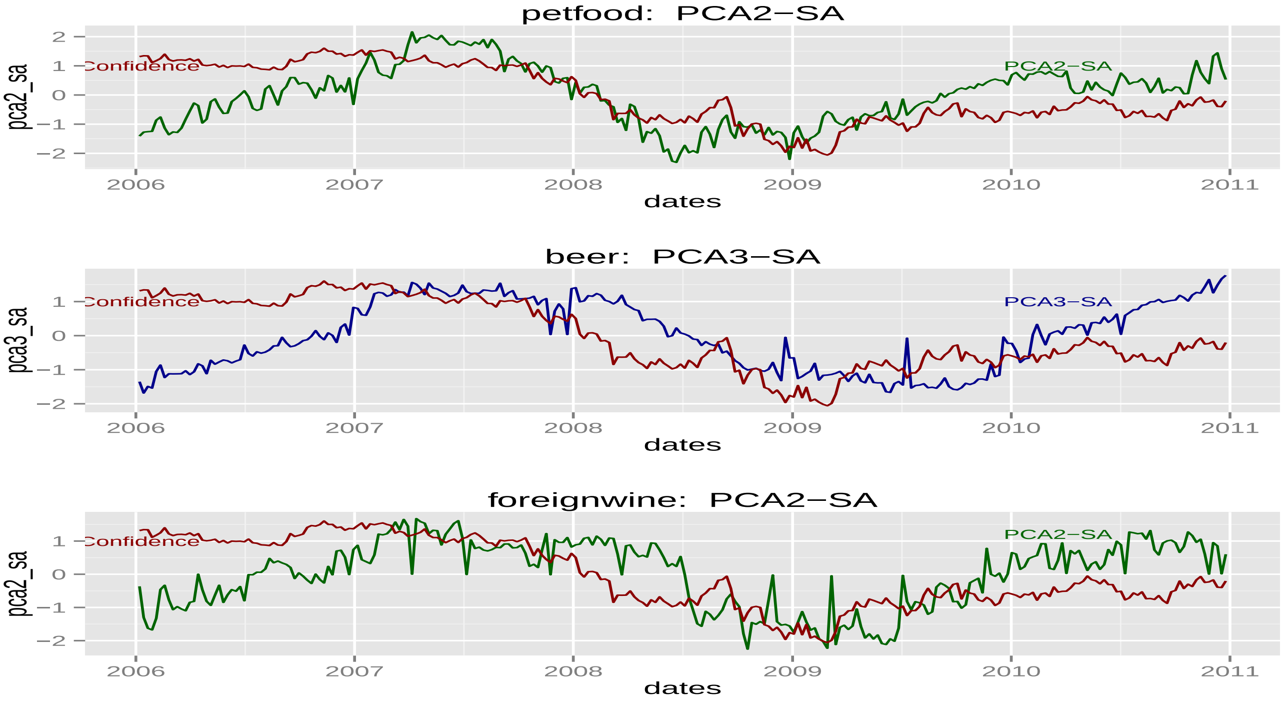
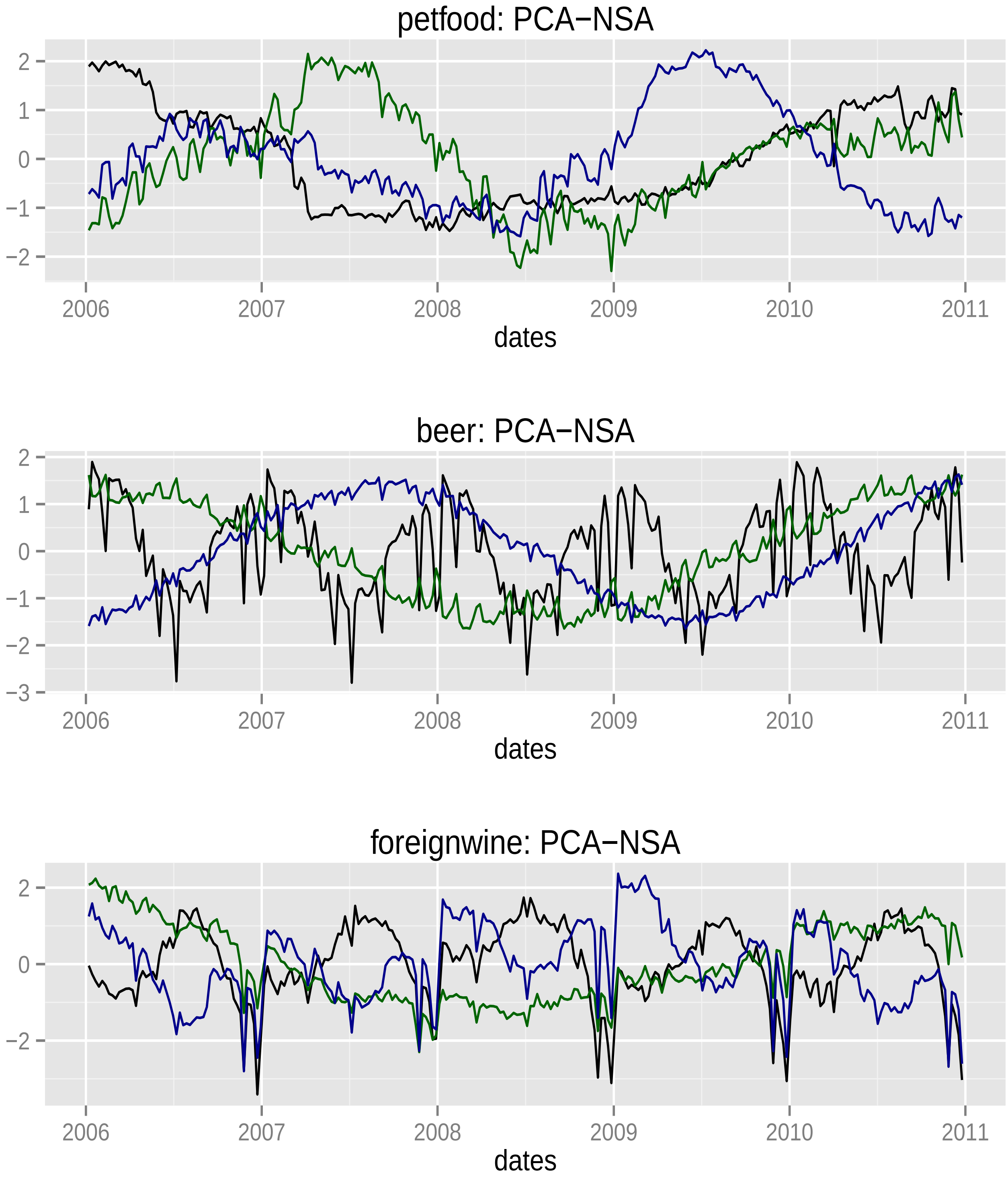


Figure 11: PCA-NSA



studying the common cyclical variations in each product, the tiny fraction may well be all that is needed. As is apparent in this exercise, the researcher has a heavy influence on what to analyze, and how. It is not a trivial task to accurately document all the steps involved. Being able to reproduce empirical results reported by other researchers is hard even when small datasets are involved. Big data make it even harder because there is more scope for subjective choices. Enforcing reproducibility of results is important, and it will not likely be an easy task.

Finally, big data can give interesting insights that may not be gleaned from conventional data. It will be useful to learn methods outside of the standard econometric toolbox as big data is likely here to stay. But while it is tempting to jump onto the big data bandwagon, one must be prepared that the learning curve can be steep, and the haystack from which to find the needle of economic insight can be huge.

# References

Achiloptas, D. 2003, Database Friendly Random Projections: Johnson-Lindenstrauss with Binary Coins, *Journal of Computer and System Sciences* **66**(4), 671–687.

Agarwal, P. S. H.-P. and Varadarajan, K. 2004, Apprixmating Extent Measures of Points, *Journal of the ACM* **51**(4), 606–635.

Ailon, N. and Chazelle, B. 2006, Approximate Nearest Neighborhood and the Fast Johnson-Lindenstrauss Transform, *Proceedings of the 38st Annual Symposium on the Theory of Computing (STOC)* pp. 557–563.

Antenucci, D., Cafarella, M., Levenstein, M., Ré, C. and Shapiro, M. D. 2014, Using Social Media to Measure Labor Market Flows, NBER Working Paper 20010.

Athey, S. 2013, How Big Data Changes Business Management, *Stanford Graduate School of Business.*

Athey, S. and Imbens, G. 2015, Machine Learning Methods for Estimating Heterogeneous Causal Effects, arXiv:1504.01132.

Belloni, A., Chernozhukov, V. and Hansen, C. 2014, High-Dimensional Methods and Inference on Structural and Treatment Effects, *Journal of Economic Perspectives* **28**(2), 29–50.

Beraja, M., Hurst, E. and Ospina, J. 2015, The Aggregate Implications of Regional Business Cycles, University of Chicago, mimeo.

Boldin, M. and Wright, J. 2015, Weather Adjusting Employment Data, Johns Hopkins University, mimeo.

Boutsidis, C., Mahoney, M. and Drineas, P. 2008, Unsupervised Feature Selection for Principal Component Analysis, KDD.

Boutsidis, C., Mahoney, M. W. and Drineas, P. 2009, An Improved Approximation Algorithm for the Column Sum Selection Problem, *Proceedings of the 20th Annual SODA*, pp. 968–977.

Brieman, L. 2001, Statistical Modeling: The Two Cultures, *Statistical Science* **16**(3), 199–215.

Broda, C., Leibtag, E. and Weinstein, D. 2009, Thr Role of Prices in Measuring the Poor's Living Standards, *Journal of Economic Pers* **23**(2), 77–97.

Cavallo, A. 2012, Online and Official Price Indexes: Measuring Argentina's Inflation, *Journal of Monetary Economics* **60**, 152–165.

Cavallo, A., Cavallo, E. and Rigobon, R. 2013, Prices and Supply Disruptions During Natural Disasters, NBER Working paper 19474.

Cha, W., Chintagunta, P. and Dhar, S. 2015, Food Purchases During the Great Recession, Kilts Booth Marketing Series, Paper 1-008.

Chevalier, J., Kashyap, A. and Rossi, P. 2003, Why Don't Prices Rise During Periods of Peak Demand? Evidence from Scanner Data, *American Economic Review* **93**(1), 15–37.

Choi, H. and Varian, H. 2012, Predicting the Present with Google Trends, *Economic Record* **88**, 2–9.

Cleveland, W. and Scott, S. 2007, Seasonal Adjustment of Weekly Time Series with Application to Unemployment Insurance Claims and Steel Production, *Journal of Official Statistics* **23**(2), 209–221.

Cleveland, W. S. 2001, Data Science: an Action Plan for Expanding the Technical Areas of the Field of Statistics, *International Statistical Review* **69**(1), 21–26.

Coibion, O., Gorodnichenko, Y. and Hong, G. 2015, The Cyclicality of Sales, Regular and Effective Prices: Business Cycle and Policy Implications, *American Economic Review* **7**, 197–232.

Deaton, A. and Ng, S. 1998, Parametric and Nonparametric Approaches to Tax Reform, *Journal of the American Statistical Association* **93**(443), 900–909.

Drineas, P., Mahoney, M., Muthukrishnan, S. and Sarlos, T. 2011, Faster Least Squares Approximation, *Numerical Mathematics* **117**, 219–249.

Drineas, P., Mahoney, M. W. and Muthukrishnan, S. 2008, Relative Error CUR Matrix Decompositions, *Siam Journal of Matrix Analysis and Applicatons* **30**, 844–811.

Du Mouchel, W., Volinsky, C., Johnson, T., Cortes, C. and Pregibon, D. 1999, Squashing Flat Files Flatter, *Proceedings of the Fifth ACM Conference on Knowledge Discovery and Data Mining* pp. 6–15.

Duhigg, C. 2-16-2012, How Companies Learn your Secrets.

Einav, L. and Levin, J. 2013, The Data Revolution and Economic Analysis, *Innovation Policy and the Economy* **NBER**, forthcoming.

Einav, L. and Levin, J. 2014, Economics in the Age of Big Data, *Science* **346**(6210), 1243089–1–6.

Fan, J., Han, F. and Liu, H. 2014, Challenges of Big Data Analysis, *National Science Review* **1**, 293–314.

Frieze, A., Kannan, R. and Vempala, S. 2004, Fast Monte Carlo Algorithms for Finding Low-Rank Approximations, *Journal of the ACM* **51**(6), 1025–1041.

Ginsberg, J., Mohebbi, M., Patel, R., Brammer, L., Smolinski, M. and Brillant, L. 2009, Detecting Influenza Epidemics Using Search Engine Query Data, *Nature* **457**, 1012–1014.

Goel, V. 8-2-2014, How Facebook Sold you Krill Oil.

Golub, G. 1965, Numerical Methods for Solving Linear Least Squares Problem, *Nuremical Mathematics* **7**, 206–216.

Granger, C. 1988, Extracting Information from Mega-Panels and High-Frequency Data, *Statistica Neerlandica* **52**(3), 258–272.

Gu, M. and Eisenstat, S. 1996, Efficient Algorithms for Computing a Strong Rank-Revealing QR Factorization, *SIAM Journal of Scientific Computing* **17**(4), 848–869.

Halko, P., Martinsson, P. G. and Tropp, J. A. 2011, Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions, *Siam Review* **53**(2), 217–288.

Handbury, J., Watanabe, T. and Weinstein, D. 2013, How Much do Official Price Indexes Tell Us About Inflation, NBER Working Paper 19504.

Harvey, A. and Koopman, S. 1993, Forecasting Hourly Electricity Demand Using Time Varying Splines, *Journal of the American Statistical Association* **88**, 1228–1236.

Harvey, A., Koopman, S. and Riana, M. 1997, The Modeling and Seasonal Adjustment of Weekly Observations, *Journal of Business and Economic Statistics* **15**, 354–368.

Johnson, W. and Lindenstauss, J. 1994, Extensions of Lipschitz Maps into a Hilbert Space, *Contemporary Mathematics.*

Jolliffe, I. 1972, Discarding Variables in a Principal Component Analysis: Artificial Data, *Applied Statistics* **21**(2), 160–173.

Koop, G. and Onorante, L. 2013, Macroeconomic Nowcasting Using Google Probabilities, University of Strathclyde.

Li, P., Hastie, T. and Church, K. 2006, Very Sparse Random Projections, *KDD* pp. 287–296.

Ma, P., Mahoney, M. W. and Yu, B. 2014, A Statistical Perspective on Algorithmic Leveraging, *Proceedings of the 31st ICML Conference*, Vol. arXiv: 1306.5362.

Madigan, D., Raghavan, N., Dumouchel, W., Nason, M., Posse, C. and Ridgeway, G. 1999, Likelihood-Based Data Squashing: A Modeling Approach to Instance Construction, *Technical report*, AT and T Labs Ressearch.

Mahoney, M. W. 2011, Randomized Algorithms for Matrices and Data, *Foundations and Trends in Machine Learning*, http://dx.doi.org/10.1561/2200000035 edn, Vol. 3:2, NOW, pp. 123–224.

Maillard, O. and Munos, R. 2012, Linear Regression with Random Projections, *Journal of Machine Learning Research* **13**, 2735–2772.

Menne, N. J., Durre, I., Vose, R., Gleason, B. and Houston, T. 2012, An Overview of the Global Historical Climatology Network-Daily Database, *Journal of Atmospheric and Oceanic Technology* **29**, 897–910.

Owen, A. 1990, Empirical Likelihood Ratio Confidence Region, *Annals of Statistics* **18**, 90–120.

Pierce, D., Grupe, M. and Cleveland, W. 1984, Seasonal Adjustment of the Weekly Monetary Aggregate: A Model Based Approach, *Journal of Business and Economic Statistics* **2**, 260–270.

Preis, T., Moat, H. S. and Stanley, H. E. 2013, Quantifying Trading Behavior in Financial Markets Using Google Trends, *Scientific Reports: Nature Publishing.*

Sarlos, T. 2006, Improved Approximation Algorithms for Large Matrices via Random Projections, *Proceedings of the 47 IEEE Symposium on Foundations of Computer Science.*

Tolentino, S. 2013, Rethinking Loylaty Programs Through Big Data.

Varian, H. R. 2014, Big Data: New Tricks for Econometrics, *Journal of Economic Perspective* **28**(2), 3–28.

Vempala, S. 2004, *Random Projection Method*, Vol. DIMACS series in Discrete Mathematics of *65*, American Mathematical Society.

Venkatasubramanian, S. and Wang, Q. 2011, The Johnson-Lindenstrauss Transform: An Empirical Study, *Proceedings of the Thirteenth Workshop on Algorithm Engineering and Experments*, pp. 148–173.

Woodruff, D. 2014, Sketching as a Tool for Numerical Linear Algebra, *Foundations and Trends in Theoretical Computer Science* **10**(1-2), 1–157.

Wright, J. 2013, Unseasonal Seasonals?, *Brookings Papers on Economic Activity* **2**, 65–110.