Big data conversion techniques including their main features and characteristics

2017 edition

STATISTICAL WORKING PAPERS

eurostat

Big data conversion techniques including their main features and characteristics

2017 edition

Manuscript completed in July 2017

Neither the European Commission nor any person acting on behalf of the Commission is responsible for the use that might be made of the following information.

Luxembourg: Publications Office of the European Union, 2017

The information and views set out in this publication are those of the authors and do not necessarily reflect the official opinion of the European Union. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein.

PDF ISBN 978-92-79-70523-6 ISSN 2315-0807 doi:10.2785/461700 KS-TC-17-003-EN-N

Abstract

Big data have high potential for nowcasting and forecasting economic variables. However, they are often unstructured so that there is a need to transform them into a limited number of time series which efficiently summarise the relevant information for nowcasting or short term forecasting the economic indicators of interest. Data structuring and conversion is a difficult task, as the researcher is called to translate the unstructured data and summarise them into a format which is both meaningful and informative for the nowcasting exercise. In this paper we consider techniques to convert unstructured big data to structured time series suitable for nowcasting purposes. We also include several empirical examples which illustrate the potential of big data in economics. Finally, we provide a practical application based on textual data analysis, where we exploit a huge set of about 3 million news articles for the construction of an economic uncertainty indicator.

Keywords: Big Data, Unstructured Data, Time Series Conversion, Data Features.

Acknowledgement:

This work has been carried out by George Kapetanios[*], Massimiliano Marcellino[†] and Fotis Papailias[‡] for Eurostat under a contract with GOPA. The Eurostat project manager was Dario Buono[**].

[*] george.kapetanios@kcl.ac.uk
[†] massimiliano.marcellino@unibocconi.it
[‡] fotis.papailias@quantf.com
[**] dario.buono@ec.europa.eu

Table of contents

List of tables

List of figures

1 Introduction

In this paper we are concerned with the task of transforming unstructured big data into a limited number of time series which efficiently summarise the relevant information for nowcasting or short term forecasting the economic indicator(s) of interest. Data structuring and conversion is a difficult task, as the researcher is called to translate the unstructured data and summarise them into a format which is both meaningful and informative for the nowcasting exercise.

The available literature on data structuring is in its early stages, with few contributions specifically focusing on this problem when using big data for economic forecasting or nowcasting. The common approach is to use summary time series indicators readily available, Google trends in particular, as highlighted in the literature. Instead, in this paper we would like to provide general rules and approaches that enable this complex data transformation and reduction, with the goal of extracting from large unstructured datasets a smaller number of time series indicators, which can then be analysed with either standard or high dimensional time series tools.

The rest of the paper is organised as follows. Section 2 presents a brief discussion of the related literature. Section 3 provides a general framework for the conversion of big data into time series format along with simulated cases. We discuss feature extraction, data mining, clustering, and random subsampling in separate subsections. Section 4, which is entirely new, illustrates various empirical big data examples, using publicly available samples. In Section 5, which is also entirely new, we illustrate big data structuring by analysing the various steps involved in the construction of a daily uncertainty indicator, potentially relevant for nowcasting several economic variables, starting from a dataset of over 3 million economic and financial articles covering a period of about 10 years. Section 6 draws the main conclusions of this paper.

2 Literature review

The literature on big data conversion techniques is very limited and mostly dependent on the specific type of big data under investigation and specific goal of the analysis. For example, let us consider a mobile phones dataset, which typically comes with the following fields: (i) the caller ID (anonymised), (ii) the caller's country and regional phone code, (iii) the receiver's country and/or regional phone code, (iv) the call duration, (v) the time of the call, and (vi) the location of the caller during the call. If our research purpose is to investigate the callers' profile across regions, which could be useful for custom/localised marketing campaigns and offers, then we must focus more on a spatial approach and summarise the big data accordingly. However, if we want to study specific types of callers' behaviour over time, we have to adopt a time series approach and structure the data using mainly the temporal dimension. Or, in a more complex setup, we might want to summarise the behaviour of callers across both geographical areas and time, which would require taking into proper consideration both the cross-sectional and the temporal dimensions.

Our approach of the literature review in this paper is divided in two parts: (i) we provide a general discussion regarding the conversion of unstructured to structured datasets, and (ii) we review the specific data conversion techniques used in the most relevant papers to our research.

2.1 Data mapping

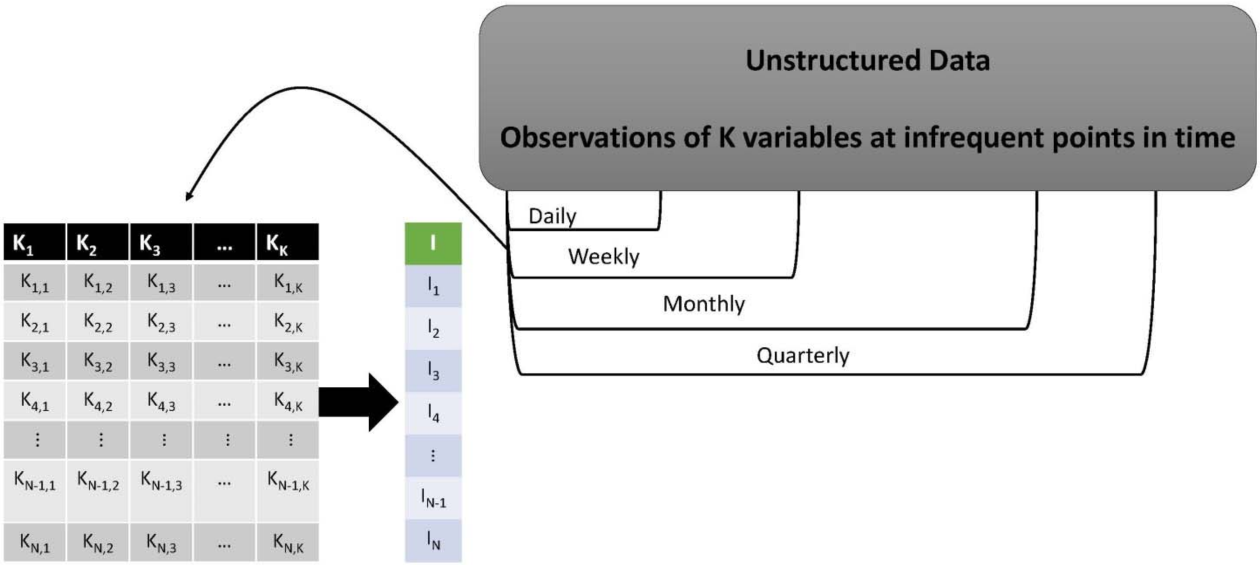
In general, we can consider four main steps in the process for mapping unstructured data to structured data; see e.g., Abdullah and Ahmad (2013). These are:

1. Extraction. Identify unstructured data and develop metadata for unstructured data.
2. Classification. Identify main data classes and categorise data according to these classes resulting in a categorisation of unstructured data.
3. Repositories Development. Preparation and development of repositories.
4. Data Mapping: Preparation of Thematic (Topic/Business, Interests/Business Subjects), Data Mapping from Repositories to Thematic. The output will be a structured dataset.

The Metadata can be created using the Dublin Core Metadata Element (DCMI, 2007) which identifies the following elements: 1. Title, 2. Explanation, 3. Date, 4. Identifier, 5. Creator, 6. Publisher, 7. Type, 8. Source, 9. Subject, 10. Contributor, 11. Format, 12. Language, 13. Relation, 14. Location, 15. Rights. The structured data is often created using the above metadata in a way to meet the needs of the applied researcher. For example, often textual data from Social networks is aggregated and then summarised in time frames selected by the user. Eurostat's ESS-MH system could be also be used in this procedure making the appropriate adjustments.

As Wu, Zhu, Wu and Ding (2014) mention, big data is characterised by the HACE theorem which is: “Big Data starts with large-volume, **h**eterogeneous, **a**utonomous sources with distributed and decentralized control, and seeks to explore **c**omplex and **e**volving relationships among data". These characteristics make it an extreme challenge to uncover useful knowledge from the big data.

Figure 1: General description of big data conversion



K_1	K_2	K_3	…	K_K
$K_{1,1}$	$K_{1,2}$	$K_{1,3}$	…	$K_{1,K}$
$K_{2,1}$	$K_{2,2}$	$K_{2,3}$	…	$K_{2,K}$
$K_{3,1}$	$K_{3,2}$	$K_{3,3}$	…	$K_{3,K}$
$K_{4,1}$	$K_{4,2}$	$K_{4,3}$	…	$K_{4,K}$
⋮	⋮	⋮	⋮	⋮
$K_{N-1,1}$	$K_{N-1,2}$	$K_{N-1,3}$	…	$K_{N-1,K}$
$K_{N,1}$	$K_{N,2}$	$K_{N,3}$	…	$K_{N,K}$

Source: Authors' calculations

The most difficult conversion problem is when the unstructured data is not characterised by a specific time frequency. Figure 1 provides an intuitive summary of the problem. Assume we have an unstructured big data of K variables with observations occurring at any point in time. The big data conversion to time series structure is twofold: (i) summarise the information of K variables across N periods of time, and (ii) summarise the K variables in one or few indicator variables. For example, we can use the simplest form of conversion which is aggregation, i.e. the summation of information over a specified time interval, e.g., at hourly, daily, weekly, monthly or less frequent time intervals. Then the big data has a structured time series dimension. Depending on the nature of our problem, we might stop at this step if we want to end up with a NxK matrix of structured data. Otherwise, we continue in the second step which is to summarise the information across the many N variables into a lower number of indicators([1]).

2.2 Recent research papers

In this subsection we review the specific data structuring and conversion techniques used in the empirical applications in the most relevant papers to our research. We do not discuss papers in the Financial Markets Data group, as the majority of them focuses on realised volatility, which is analysed in details in the next section. This review is also useful for future projects, as often data structuring is jointly considered with signal extraction and pre-treatment.

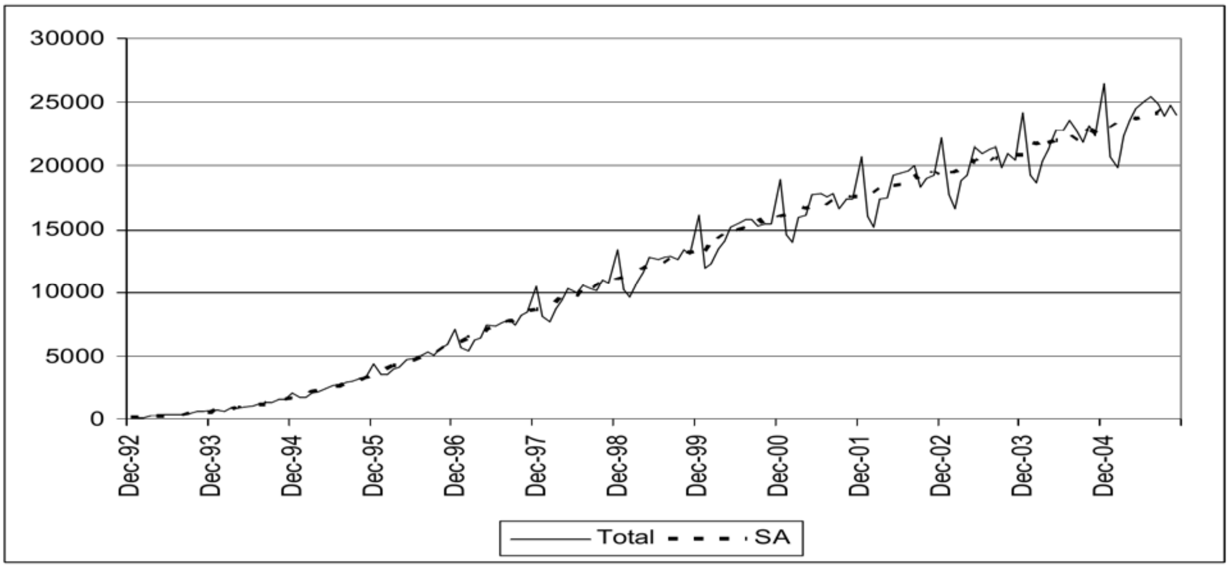
2.2.1 Electronic payments data

Galbraith and Tkacz (2007) summarize the available big data by taking the sum of all debit card transactions in Canada, aggregated on a monthly basis. Details regarding pre-treatment and/or cleaning of the data are not discussed, given that the dataset is provided by the Interac Organisation (third-party). The main evident feature in the monthly time series is the seasonal pattern; see Figure 2. The authors deseasonalise the series using X11-ARIMA([2]). Galbraith and Tkacz (2011) use the same dataset comparing daily and monthly aggregates, reporting similar stylised facts.

([1]) Einav and Levin (2013) also identify the challenges faced by economics in big data conversion. Traditionally, economic researchers had to deal with NxK matrices. Nowadays they must figure out how to organise a sequence of events, with no further structure, reducing its dimensionality and assessing the way the imposed structure matters. This is becoming a very common challenge in empirical research and our aim is this paper to contribute to this literature.

([2]) See Kapetanios et al. (2016) for a discussion regarding X11-ARIMA.

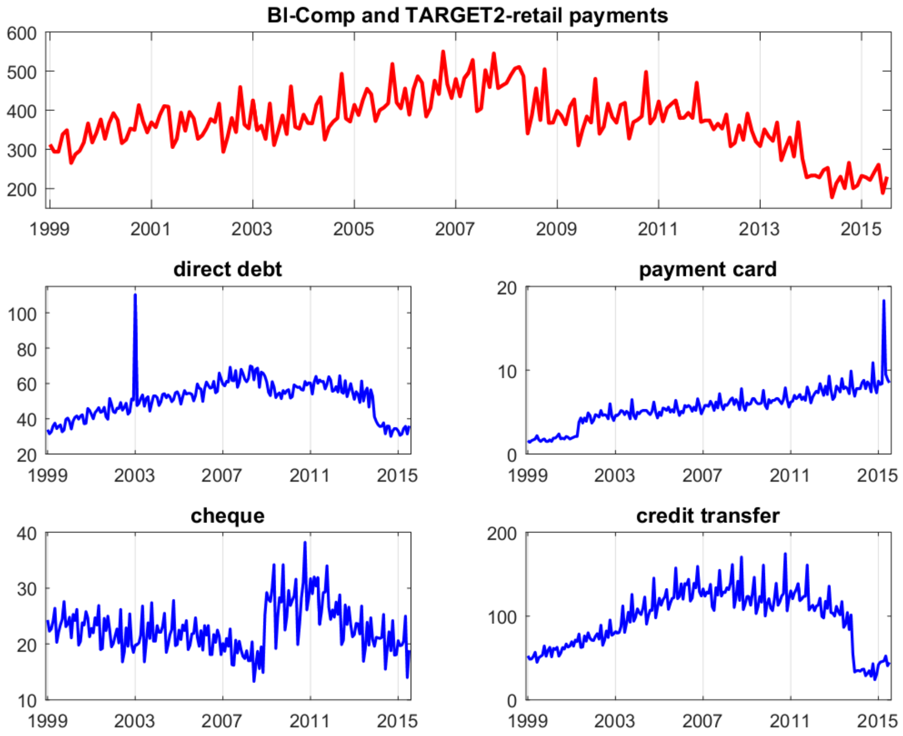
Figure 2: Galbraith and Tkacz (2007) data example



Source: Galbraith and Tkacz (2007)

Similar seasonal patterns also emerge in Esteves (2009) and Carlsen and Storgaard (2010), however in both papers the authors do not provide much information regarding the aggregation and conversion techniques. Aprigliano, Ardizzi and Monteforte (2016) also use the aggregated value of transactions without specifying pre-treatment or other prior cleaning steps, see Figure 3.

Figure 3: Aprigliano, Ardizzi and Monteforte (2016) data example

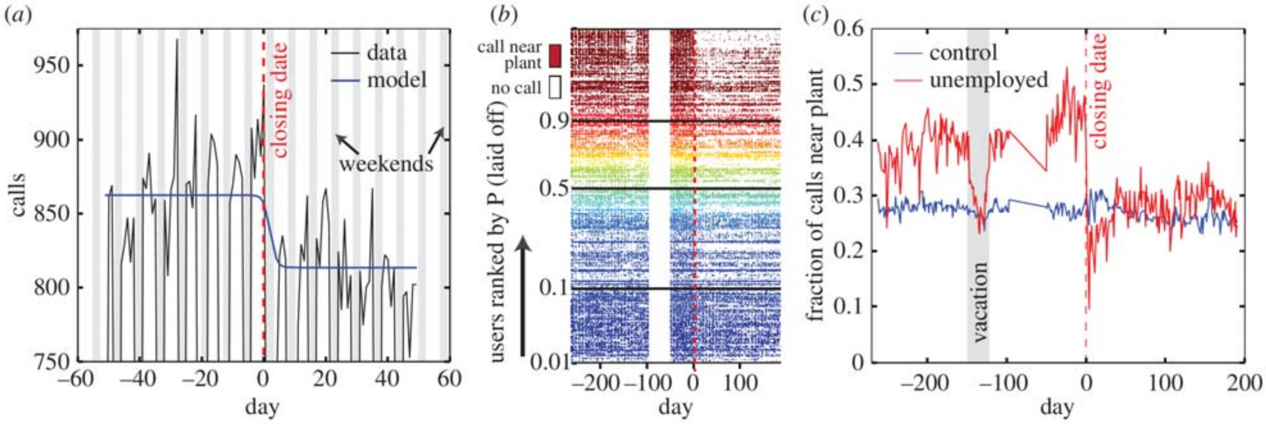


Source: Aprigliano, Ardizzi and Monteforte (2016)

2.2.2 Mobile phone usage data

Almost all research studies on mobile phone usage data map the data from a geographical/spatial perspective and do not convert it to time series. However, Toole, Lin, Muehlegger, Shoag, Gonzalez, and Lazer (2015) use also the temporal dimension, related to mobile phone activity. In particular, they construct a time series of total aggregate call volume that, as in the previous subsection, is obtained by simple linear aggregation; see Figure 4.

Figure 4: Toole et al. (2015) data example



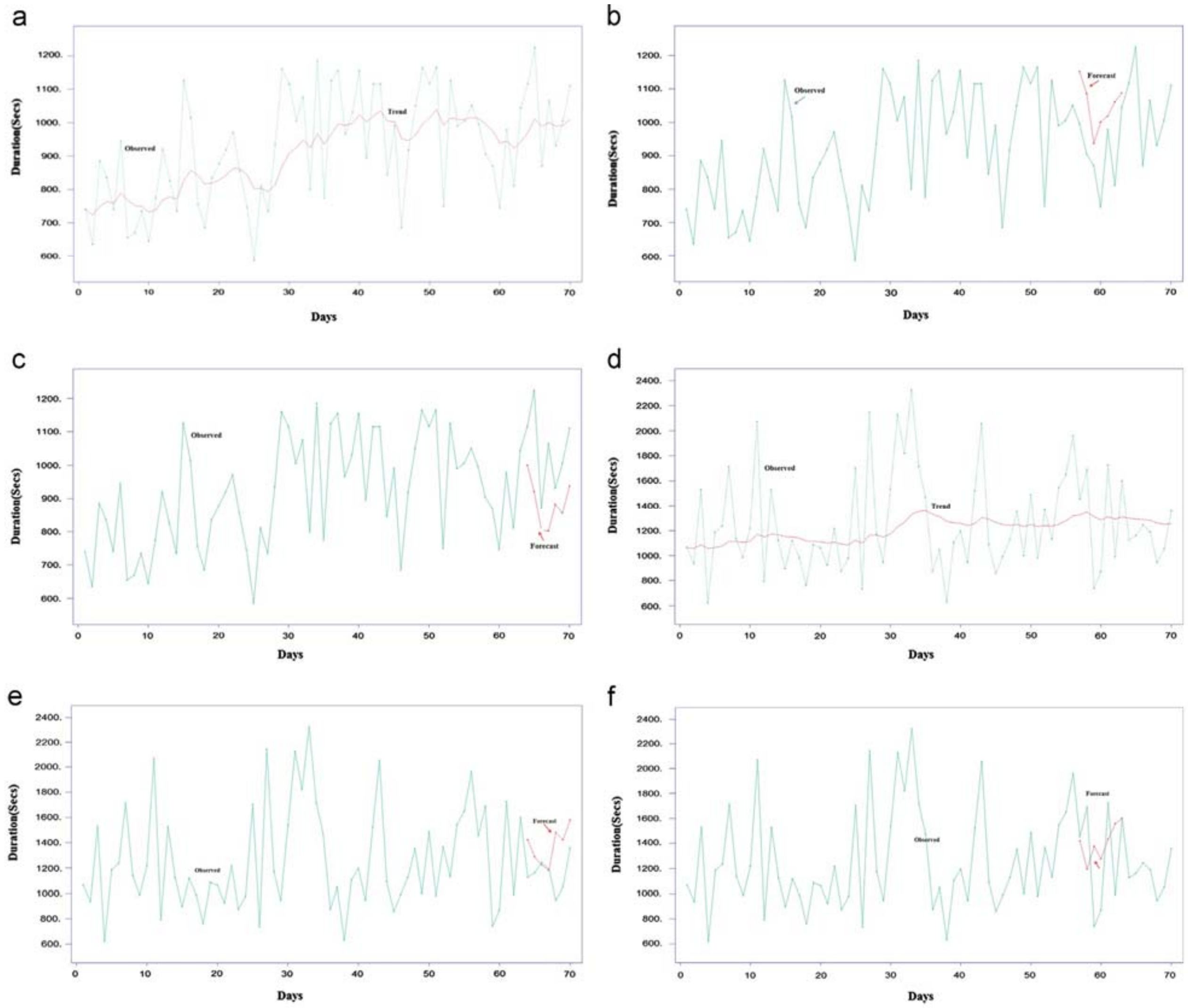
Source: Toole et al. (2015)

De Meersman, Seynaeve, Debusschere, Lusyne, Dewitte, Baeyens, Wirthmann, Demunter, Reis and Reuter (2016) assess the quality of mobile phone data as a mean to improve population statistics. They use aggregated data, “possibly the simple summation across time", and report encouraging results on the usability of mobile phone data to complement traditional census statistics.

2.2.3 Sensor data

The literature on sensor data documents mostly the conversion of big data into geographical/spatial observations. For example, Suryadevara, Mukhopadhyay, Wang and Rayudu (2013) investigate the behaviour of an elderly person using various sensors in her home appliances. They analyse toilet usage, chair usage and the usage of other appliances, aggregating data received over the day; see Figure 5.

Figure 5: Suryadevara et al. (2013) data example



Source: Suryadevara et al. (2013)

Fernandez-Ares, Mora, Arenas, de las Cuevas, Garcia-Sanchez, Romero, Rivas, Castillo and Merelo (2016) focus on traffic nowcasting. In this case a time series can be constructed by linear aggregation of the number of cars in circulation according to the data transmitted from the sensors (possibly disaggregated by geographical areas to obtain a panel structure, and at the temporal frequency of interest, e.g., at 5 minute intervals).

2.2.4 Satellite images data

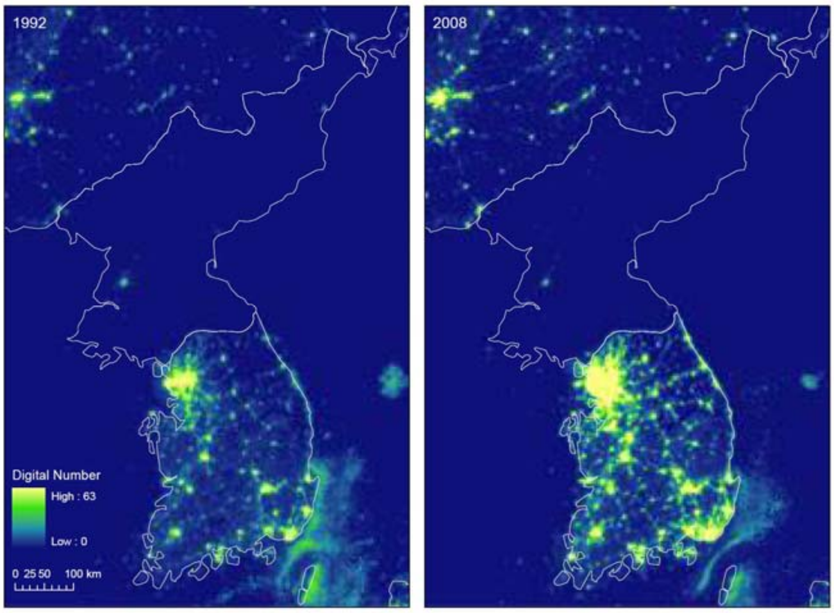
When using night lights data, most of the papers focus on explaining or forecasting GDP growth on an annual basis. Henderson and Storeygard (2011, 2012), Pinkovskiy (2013), Mellander, Lobo, Stolarick and Matheson (2015) and others([3]) use the National Oceanic and Atmospheric Administration's (NOAA) National Geophysical Data Center (NGDC) source. In these studies, the overall daily data is disaggregated into specific countries, with different colours indicating the intensity of lumination; see Figure 6. Then, e.g. Henderson and Storeygard (2011), for a given satellite year, calculate the average intensity of lighting. Repeating this procedure for each year yields an average luminosity time series for each country.

([3]) See Lowe (2014) for a comprehensive guide in satellite image data manipulation.

2.2.5 Price data

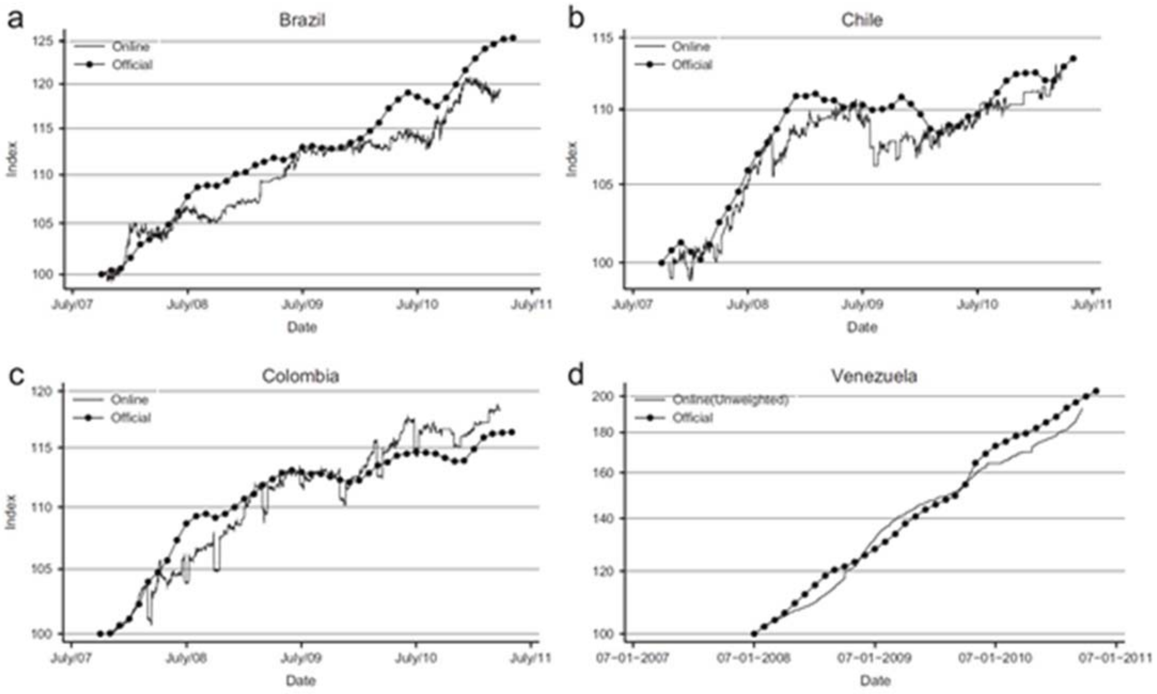
Next, we turn our attention to scanner prices and online prices data. Between the two, online prices data is more frequently used in more recent applications, as prices observation can be collected daily, whereas in scanner data the products are scanned only at pre-specified times. Therefore, we focus more on applications based on online prices data. Within this set, Cavallo (2013) is perhaps the most representative example. As we discuss in more details in the next section, daily scanner and online prices data on specific products are first cross-sectionally aggregated, as when constructing a price index, and then they are temporally aggregated into monthly time series, resembling official consumer price indexes, see Figure 7.

Figure 6: Henderson, Storeygard and Weil (2012) data luminosity example



Source: Henderson, Storeygard and Weil (2012)

Figure 7: Cavallo (2013) example



Source: Cavallo (2013)

2.2.6 Textual data

The last big data type we look into is textual data. Textual data can be in the form of news articles, social media feeds, e.g. Twitter, online search data, etc. Gentzkow, Kelly and Taddy (2016) provide a detailed discussion about textual data([4]). The first step when working with textual data is to transform it into numerical format by extracting specific features; e.g. particular keywords, word sequences, letters, etc. Once features are extracted, then a time series can be constructed using an aggregation function for the specific time frequency of interest, e.g., counting the total number the specific keyword of interest appears in a day or in a week. Textual data has been used in the construction of sentiment indicators([5]), as we discuss in more details in the next sections, see in particular Section 5.

For online search data, the most common example is Google trends, which are based on the count of the number of queries for specific keywords (treated as textual features), temporally aggregated at the weekly frequency. This facilitates the nowcasting procedure, as the researcher, after selecting the specific keywords of interest, often does not have to further edit the resulting Google trends. However, some additional data cleaning and preparation might still be necessary and/or improve the nowcasting results, e.g., the removal of outliers, temporal patterns, or the extraction of a clearer signal.

([4]) Also see Gandomi and Haider (2015) for a brief summary of textual data analytics.

([5]) See, e.g., Baker, Bloom and Davis (2016) among others.

3 A general data conversion framework for unstructured numerical Big Data

In this paper we introduce a generalised data conversion framework for big data in numerical format. The framework also applies to categorical big data, e.g. Twitter feeds, news, etc., after extracting a numerical feature in a first step. For example, consider the case of Twitter data. A feature would be to count the number of times a chosen keyword appears in Twitter feeds every 30 seconds during a business day.

This section aims to offer a general unstructured data conversion framework. It must be noted that data structuring can be made in different ways producing different results. The ways in which the goal of structuring can be achieved depends on the targeted phenomenon, the granularity of the phenomenon and the characteristics of the unstructured data. Also, the dimensions of the structured dataset can have relevant impact on the modelling process. For example, if the output of the structuring process is one or a very few time-series, then there will be not necessarily the need of specific techniques dealing with large and sparse dataset and the focus will be mostly on mixed frequency approaches. The next section provides specific examples for big data types which are variations of the general data conversion framework presented here.

3.1 Conceptual setting

We start by considering an unstructured dataset which consists of N events, each described by a vector of the form $\mathrm{yi} = (\mathrm{y1,i,\ldots,ym,i,ti})' = (\tilde{\mathrm{y}}_{\mathrm{i}}', \mathrm{ti})'$, $\mathrm{i} = 1,\ldots,\mathrm{N}$, where N is potentially very large. The vector $\tilde{\mathrm{y}}_{\mathrm{i}}'$ contains information on the event, while ti denotes the time the event has occurred, and is referred to as a time stamp. This definition of an event in terms of a vector of characteristics and time stamp is very general. It is difficult to envisage any big dataset with a temporal dimension that cannot be accommodated by this definition. Time is defined continuously in an interval $(0,\mathrm{T}]$. Our aim is to describe a mapping between these events and potential time series defined in discrete time, $\mathrm{t} = 1,\ldots,\mathrm{T}$, which can then be used for nowcasting or other econometric analyses.

We first need to define a mapping between the set $\tau = \{\mathrm{t_i}\}_{\mathrm{i}=1}^{\mathrm{N}}$and time series observations. That is, a set function given by $\tau_{\mathrm{t}} = \mathrm{f}(\tau,\mathrm{t})$ where τ_{t} is a subset of τ containing the event time stamps that relate to time period t. The obvious mapping is for τ_{t} to contain the events whose time stamp satisfies $\mathrm{t} - 1 < \mathrm{ti} \leq \mathrm{t}$ but more complex setups, e.g. with lags, are possible. For example, it may be that τ_{t} contains the events whose time stamp satisfies $\mathrm{t} - \mathrm{p} < \mathrm{ti} \leq \mathrm{t} - \mathrm{p} + 1$.

Then, we can define a time series as

$$x_t = \sum_{t_i \in \tau_t} g_t(y_i, \gamma) \qquad (1)$$

where $\mathrm{gt}\,(\mathrm{yi},\gamma)$ is a function possibly depending on t (to reduce the notational burden we sometimes suppress the t subscript), and parameterized by γ, mapping information on the event (contained in $\tilde{\mathrm{y}}_{\mathrm{i}}'$) into a single number. This is a very general formulation and our discussion below provides ways in which we can better interpret its applicability. Before proceeding it is important to note that the only restriction imposed here is a form of linear separability across events. The alternative would be a formulation of the type $\mathrm{xt} = \mathrm{gt}\,(\mathrm{y1},\ldots,\mathrm{y\tau t}\,;\,\gamma)$. This would provide the ability to aggregate events in a nonlinear fashion, but we feel this might be unwieldy in practice. We also note that there is the potential for different time scales to be used in defining the mapping from events to time series. In particular, we can use the above mapping framework to construct, for example, both quarterly and monthly time series which can then be used by considering mixed frequency methods. Finally we note that missing fields or elements in yt can simply be treated by removing the events or, if the incidence of missing values is great, by setting indicator functions, associated with selecting particular features, to zero.

3.2 Aggregation

We suggest to consider g functions of the type:

$$g_t(y_i, \gamma) = \sum_{j=1}^{m} w_{tj} g_{tj}(y_{j,i}, \gamma_j), \qquad (2)$$

where the $wtjs$ are weights. For example, the function could be $gj\,(yj, i, \gamma j\,) = 1$, or $gj\,(yj, i, \gamma j\,) = y_{j,i}^{\gamma_j}$ or $gj\,(yj, i, \gamma j\,) = yj, i\ I(|yj, i| > \gamma j)$, with $(w_{t1}, w_{t2}, \dots, w_{tm})$ depending on t, and equal to, e.g., $(1, 1, \dots, 1)$ or $(1/m, 1/m, \dots, 1/m)$ or even $w_{tj} = \frac{w_{tj}}{\tau_t}$ thereby providing a useful normalisation for the summation.

The function in (2) is a linear transformation of (possibly non-linear) gtj functions. As an alternative, a non-linear transformation could be considered. Moreover, the wtj weights could be also optimally computed given a certain loss function.

An interesting possibility is to cast in this set-up the construction of internet search-based indicators, such as Google trends. In this context, $(y1, i, \dots, ym, i) = (y1, \dots, ym)$ denotes the numerical representation of alphanumeric sequences that rep- resent generic internet searches, for example $y1$ corresponds to "recession" and $y2$ to "income" (so $m = 2$ in this example)([6]). Then, we can define

$$g\,(y_i, \gamma) = I\,(y_i \in \mathcal{G}_{t_i}(\gamma)), \qquad (3)$$

where $\mathcal{G}ti\,(\gamma)$ denotes (possibly a subset of) all Google searches during period ti (transformed into numerical representation), and may depend on the parameter vector γ (for example, γ can select searches for only a specific country or in a specific language), so that the function $g\,(yi, \gamma)$ counts how many times yi (a search for both "recession" and "income") happened in period ti. Computing $g\,(yi, \gamma)$ for $i = 1, \dots, N$ and plugging the results into (1) returns a "Google trend" time series for period $= 1, 2, \dots, T$. Naturally, if $m = 1$ only single keyword searches are considered, so we would get two separate Google trends for "recession" and "income".

As another example, consider credit card data. In this case, each element of $(y1, i, \dots, ym, i)$ can be either a binary number denoting whether there was a transaction at unit $j = 1, \dots, m$ in period ti, or a continuous number representing the monetary amount of the transaction. If we define $gj\,(yj, i, \gamma j\,) = yj, i$ and $wj = 1$, then xt measures either the total number of transactions in period t, or their total monetary amount. The parameters γj could be used, for example, to select specific sectors or units, while the weights wj could be used to obtain also (possibly) weighted averages rather than totals only.

As yet another example, consider scanner price data for food products. In this case, the elements of $(y1, i, \dots, ym, i)$ represent the prices of products $j = 1, \dots, m$ in period ti. The prices can be collected at different (possibly online) retailers, which will differ for (possibly slightly) different collection times ti. Defining $gj\,(yj, i, \gamma j\,) = yj, i$ and wj the weight for product j, then xt can be used to construct a food price index in period t (or a real time index when using $g\,(yi, \gamma)$). Again, the parameters γj could be used to select specific units or subsectors, while the weights wj could be used to assess the effects on the price index of different weighting.

The function $g\,(yi, \gamma)$ is defined by the researcher and/or by the research question, but it can only be computed if access to the underlying big data is granted. Continuing the Google example, the search keywords are defined by the researcher, but the $g\,(yi, \gamma)$ can be only computed and aggregated by Google, as the owner of the search

[6] It is important to note that the list of key words that should be entered into Google Trends is a matter for consideration. However, in our view the wider the list the better, since our recommendation is that at a later stage all these variables constructed via Google Trends (or other Google facilities such as Google Correlate) can be analysed in a regression setting using various data rich methodologies. As a result one can design a wide variety of relevant keywords (the use of a dictionary or a thesaurus may be useful here, together with an automatic translation service such as Google translate) and these can then be used on their own or combined through the use of logical relationships based on "or" or "and" to provide the final list.

data. Same for the credit card and scanner price examples.

Once parameterised by, e.g., setting a subset of w_j to zero and choosing the functional form for g_j , the parameters , $w_j, \gamma_j, j = 1, \ldots, m$ can be either fixed a priori or calibrated. The most obvious approach is to consider many different events $\tilde{y}_i'$ and/or functions g, by, e.g., defining a grid for γ and w, and so obtain a large number of time series indicators, which can then be handled by appropriate econometric methods.

An interesting alternative is to pin down the function g by considering which choice of g has the best forecasting ability for the target variable when used on its own or, potentially, in combination with a lag of the target variable.

3.3 Features extraction

Apart from the above function, we can extract features from the distribution of the events which occur at time t. Depending on the nature of the underlying data, we could extract the variance, various percentiles, the skewness and kurtosis, or even higher moments, and then use the resulting time series of big data features.

Formally, and assuming that N events occur at a particular time period, t, we can extract the features using:

Variance:
$$g_t(y_i) = \frac{1}{N}\sum_{j=1}^{N}(y_{it} - \overline{y}_t)^2 \tag{4}$$

P-Percentile:
$$g_t(y_i) = y_{\left[\frac{P}{100}N\right]t}, \text{ where } [.] \text{ denotes ordinal ranking,} \tag{5}$$

Skewness:
$$g_t(y_i) = \frac{\frac{1}{N}\sum_{j=1}^{N}(y_{it}-\overline{y}_t)^3}{\left[\frac{1}{N-1}\sum_{j=1}^{N}(y_{it}-\overline{y}_t)^2\right]^{3/2}}, \tag{6}$$

Kurtosis:
$$g_t(y_i) = \frac{\frac{1}{N}\sum_{j=1}^{N}(y_{it}-\overline{y}_t)^4}{\left[\frac{1}{N}\sum_{j=1}^{N}(y_{it}-\overline{y}_t)^2\right]^{2}} - 3, \tag{7}$$

where $\overline{y}_t$ denotes the sample mean of N events at time t.

These measures are rather intuitive, with the specific choice depending on the nature of the big data and the purpose of the exercise. For example, in the case of the electronic payment data, we mentioned that typically only average payments over a certain time interval are considered. However, a measure of the dispersion of the payments around the average, or low and high percentiles, could also provide useful information for nowcasting variables such as retail sales. As another example, one could complement an economic uncertainty indicator with a measure of dispersion in the forecasts of professional forecasters; see, e.g. D'Amico and Orphanides (2008), Baker, Bloom and Davis (2016), among others. This is a case of big data if we think that we have a large sample of professional forecasters who provide GDP nowcasts every month.

Another interesting case where other summary measures besides the average of the big data can be relevant is when a very large set of alternative nowcasts are available, each coming from different models and/or indicators. For example, considering all the possible nowcasting models based on a set of N indicators yields a total of $2N$ models (already more than one million for $N = 20$). If we further assume that each indicator can enter the model with up to p lags, the number of models further increases to $2Np$. A common approach would be to combine all the resulting nowcasts, but looking at their dispersion and/or at specific percentiles is also informative, as it provides a measure of model uncertainty, i.e., small dispersion means that the majority of the models point to a similar nowcast.

3.4 Data mining

Another approach to reduce dimensionality is data mining and, in particular, clustering. To keep things simple and intuitive, we illustrate this approach by means of a basic clustering method, which is the $k - \text{means}$ clustering.

$k - \text{means}$ clustering aims to partition N observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. Given a set of observations, $(y1, \ldots, yi)$, $k - \text{means}$ clustering aims to split the data into k sets, $(s1, \ldots, sk)$, so as to minimise the within-cluster sum of squares which is the sum of distance functions of each point in the cluster to the K center. The relevant objective function is:

$$\mathrm{arg}_{s}\mathrm{min} \sum_{j=1}^{k} \sum_{y \in S_i} \left\| y - \mu_j \right\|^2, \qquad (8)$$

where μj is the mean of points in Sj . This will return k centers where $\mathrm{k} < \mathrm{N}$. In this way, the researcher succeeds in the reduction of the number of events to be considered. This method is particularly useful when similar observations repeatedly appear in the sample. In this case simple aggregation could be suboptimal, as it could hide substantial heterogeneity.

An important issue is the selection of the number of clusters, k. The elbow and the $\mathrm{silhouette}$ methods are two simple procedures to select the optimal number of clusters. The elbow method suggests to choose the number of clusters such that adding another cluster does not increase the percentage of explained variance, i.e. it reaches a plateau.

The $\mathrm{silhouette}$ method is more related to the Partitioning Around Medoids (PAM) algorithm. In contrast to $\mathrm{k} - \mathrm{means}$, which centers at the mean of the cluster, a medoid partitioning method centers at the most representative datapoint.

Once the clusters have been established, the $\mathrm{silhouette}$ value measures how similar an object is to its own cluster compared to other clusters. The silhouette ranges from -1 to 1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. If most objects have a high value, then the clustering configuration is appropriate.

Another method for the selection of k, which could also be used without the need to pre-specify the maximum number of clusters, is the Affinity Propagation Clustering (APC) of Frey and Dueck (2007). As in the k-medoids, APC finds “exemplars”, members of the input set which are representative of the clusters. Following Frey and Dueck (2007), APC takes as input a collection of real-valued similarities between data points, where the similarity $s(i,k)$ indicates how well the data point with index k is suited to be the exemplar for data point i. In the application we present later on, we use the Euclidean distance, i.e. each similarity is set to a negative squared error: for points xi and xk the similarity is given by $s(i,k) = -\|xi - xk\|2$. However, other functions can also be considered.

As mentioned, APC does not require the researcher to set the number of clusters. Instead, APC takes as input a real number $\mathrm{s(k,k)}$ for each data point so that data points with larger values are more likely to be chosen as exemplars. These values are called “preferences”. APC compares diff t points in order to identify their “relationship” or “exchanged messages”. The responsibility matrix, R, takes values $\mathrm{r(i,k)}$, which quantify how well xk serves as the exemplar for xi related to other examplars candidates. The “availability” matrix, A, takes values $\mathrm{a(i,k)}$, which rep- resent how appropriate it is for xi to pick xk as its exemplar, i.e., the response of xi to xk . Then, the following steps are performed iteratively:

1. Responsibility updates: $r(i,k) \leftarrow s(i,k) - max k' \neq k\{a(i,k') + s(i,k')\}$.
2. Availability is updated as

$$a(i,k) \leftarrow min\left(0, r(k,k) + \sum_{i' \notin \{i,k\}} max\left(0, r(i',k)\right)\right) \text{ for } i \neq k \text{ and,} \qquad (9)$$

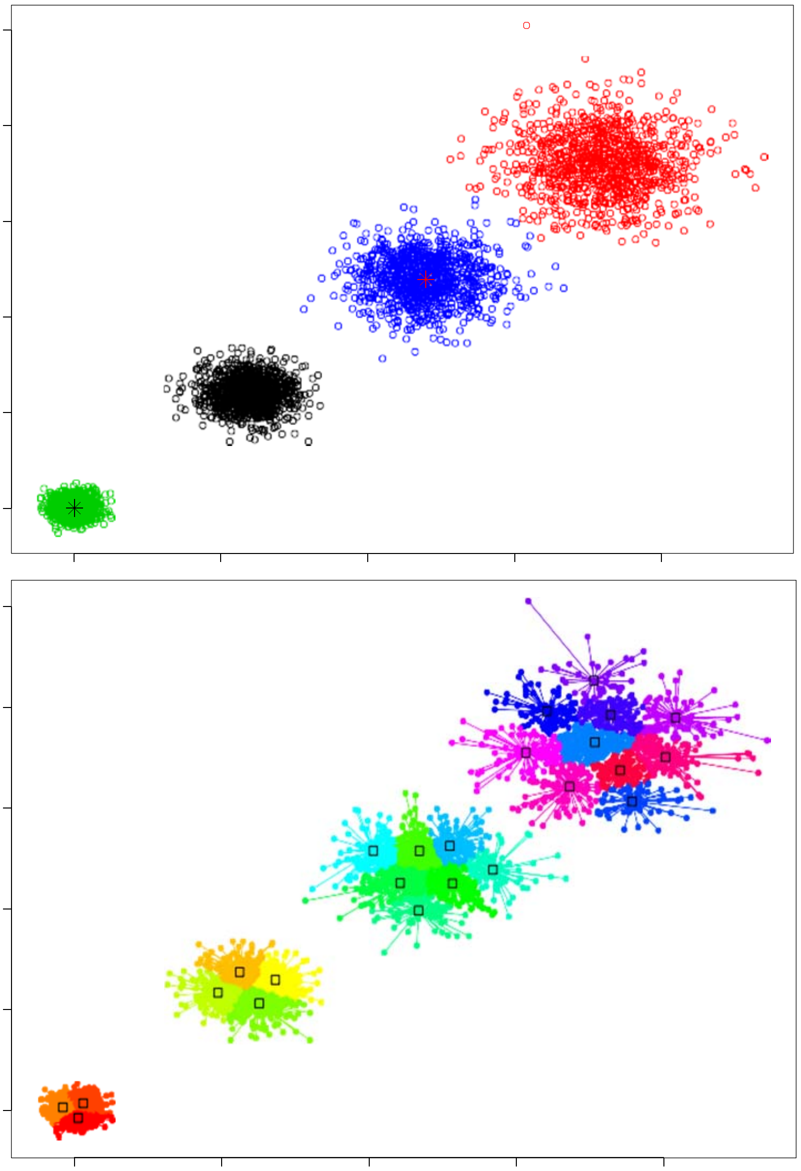
$$a(k,k) \leftarrow \sum_{i' \neq k} max\left(0, r(i',k)\right) \qquad (10)$$

The iterations are performed until either the cluster boundaries remain unchanged after some iterations or when the maximum number of iterations is reached.

To the best of our knowledge, data mining techniques have not been used in the conversion of unstructured data to time series. They have been mainly used in text analytics where the time dimension is not always of interest, see for example Gandomi and Haiver (2015) among others. A possible application of clustering methods in big data conversion could be in geographical or spatial data. As we show in the next section, clustering techniques could be employed when analysing GPS data. Focusing on a particular time frequency, say a day, we can cluster the position of an individual using the GPS coordinates and weight the cluster centres (or exemplars) by the amount of time spent in that neighborhood. Then, we can match the cluster centres or exemplars to a list of known GPS coordinates and locate how much time an individual spends in retail stores, services, commuting, etc. Repeating this procedure for a number of individuals, either aggregating them or averaging them out, we can end up with a time series which might be useful in retail sales nowcasting, services use, etc.

As an illustration of clustering, we generate four clusters of 1,000 events observed at a particular point in time. Figure 8: k-means and APC simulated example shows the centres and exemplars estimation using $\mathrm{k-means}$ and APC. We generate cluster drawing from the normal distribution with $\mu\mathrm{k} = \{0, 6, 12, 18\}$ and $\sigma\mathrm{k} = \{0.4, 0.8, 1.2, 1.6\}$ for each k cluster. The elbow method correctly identifies the use of four clusters. We see that APC identifies exemplars even within clusters which can become more informative. In both cases, we start with 4,000 observations and obtain 4 centres and 24 exemplars for $\mathrm{k-means}$ and APC respectively. These observations could now be further used in the conversion of the big data in time series when the above procedure is repeated for each time period.

Figure 8: k-means and APC simulated example



Source: Authors' calculations based on simulated data

3.4.1 Random subsampling

The above time series conversion techniques are expected to work well in applications where the researcher is interested in summaries and/or features of the data and where the size of the dataset allows the required computations. However, how should we approach cases where data is huge? For example, consider datasets which are measured in terabytes, petabytes, exabytes, zettabytes or even yottabytes. Can these conversion techniques still be applied? In cases where even database managers struggle to work well with scientific software to construct structured time series, we can adopt the random subsampling approach; see Ng (2016) and references therein.

In this approach random subsamples of the data are selected to create a new big dataset which is significantly smaller. For example, we could select observations of random seconds in an hour, or random hours in a day or even random days in a month. Then, the conversion techniques as described above can be applied and continue in the standard way. In this cost-efficient way, the researcher can randomly extract a large number of subsamples, which consequently leads to a large number of structured time series. At this point, an average of the resulting time series can be used or principal components can be extracted. The issue here is mostly one of data management and less of econometric difficulty, since a number of methods we have discussed previously stay, in principle, valid for any dataset size while the informational loss associated with subsampling has an unknown cost.

Let us consider for example a random projection that takes a set of n points in $\mathbb{R}\mathrm{d}$ and projects them to a set of n points in $\mathbb{R}\mathrm{k}$ with $\mathrm{k} << \mathrm{d}$. A random projection like this would be meaningful if the set of points in $\mathbb{R}\mathrm{k}$ preserves the structure of the original data. Johnson and Lindenstauss (1994) suggest that a set of $(\mathrm{u1}, \mathrm{u2}, \ldots, \mathrm{un})$ points in $\mathbb{R}\mathrm{d}$ can be projected down to $(\mathrm{v1}, \mathrm{v2}, \ldots, \mathrm{vn})$ points in $\mathbb{R}\mathrm{k}$ such that for any $\epsilon \in (0, 1/2)$ and $\mathrm{k} \geq \mathrm{k}_0 = \mathrm{O}(\log \mathrm{n}/\epsilon^2)$,

$$(1-\epsilon)\left\|u_i - u_j\right\|^2 \leq \left\|v_i - v_j\right\|^2 \leq (1+\epsilon)\left\|u_i - u_j\right\|^2, \tag{11}$$

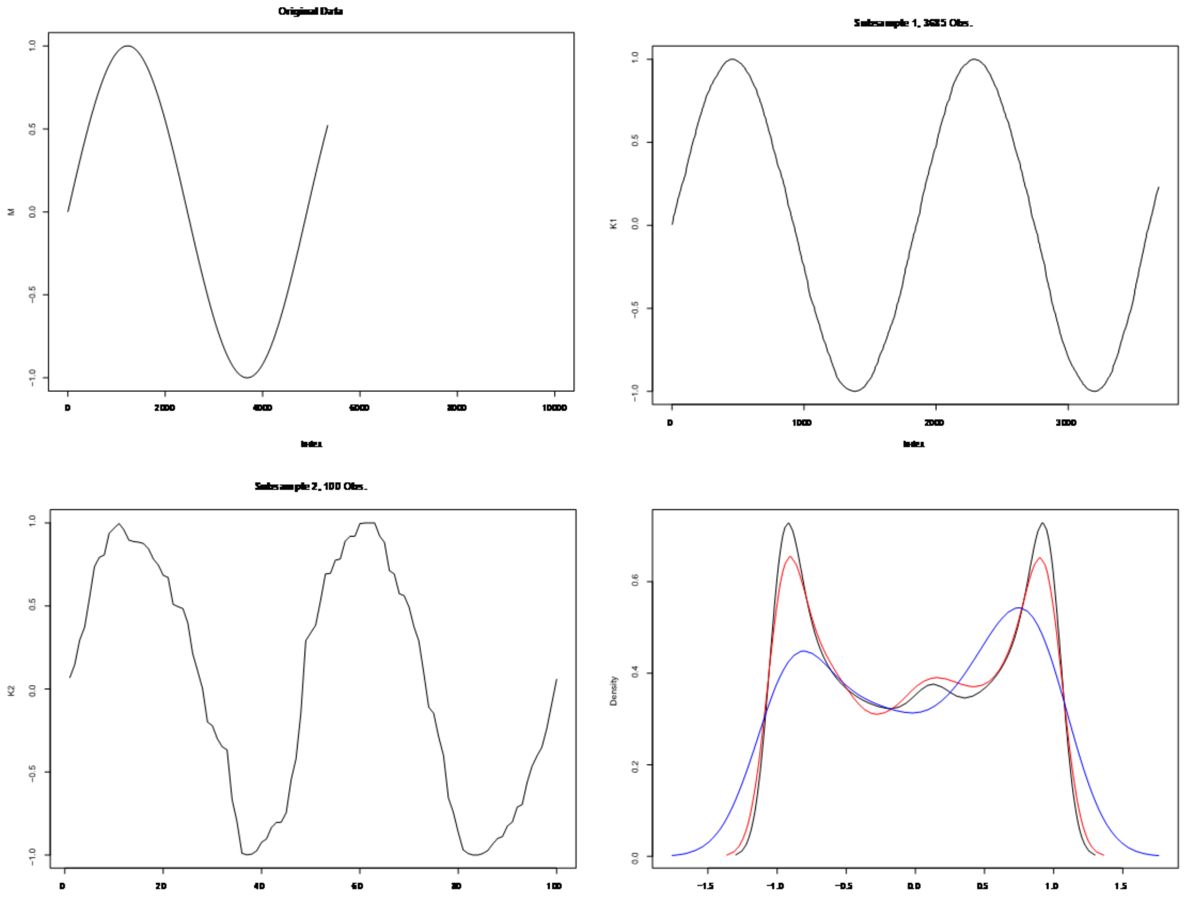
i.e., random projections generate small distortions in terms of Euclidean distance between points. The above implies that high dimensional problems can be solved more efficiently if we fi translate them in a lower dimensional space with k columns; notice that k depends on n but not d. There are many implementations of the above transformation; see Venkatasubramanian and Wang (2011) for a review. Ailon and Chazelle (2006) and Sarlos (2006) suggest to use the Fast Johnson Lindenstauss Transform (FJLT) where $\mathrm{Bk} = \mathrm{AR}$ with $\mathrm{R} = \mathrm{DHS}$ and,

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

The Hadamard transform destroys the non-uniform structure of the data. Fol- lowing Ng (2016), it can be thought of as a real-valued version of the complex Fourier transform which orthogonalises the data. Then, the orthogonalised data is re-randomised by another sparse matrix, D. The benchmark residual error is usually the best low rank approximation of A.

To illustrate the applicability of JL subsampling we generate a sample of 10,000 observations which follows the sine trigonometric function in order to ensure that data features are preserved. Then, we create two random subsamples: (i) one using the number of events needed to achieve a distortion of $\varepsilon = 0.1$, and (ii) one using a sample of 100, i.e. using 1% of the original data. The JLT indicates that the number of selected events to achieve a specific distortion is $[(4 \times \log \mathrm{n})/\varepsilon^2]$. Therefore, for $\mathrm{n} = 10{,}000$ and $\varepsilon = 0.1$ this is equal to 3685, i.e. 36.85% of the original data. Figure 9 and Table 1 compare the original data and the two subsamples. We clearly see that random subsampling can be very helpful in large datasets as the main characteristics of the original data are preserved and, as seen in the density plot, the whole distribution is well approximated.

Figure 9: JLT random subsampling examples



Note: The red line is the density of the 3685 obs. sample. The blue line is the density of the 100 obs. sample.

Source: Authors' calculations based on simulated data

Table 1: Comparing statistics of the original sample and the two random subsamples

	Original Data	Subsample 1	Subsample 2
Average	0.002114	0.001182	0.039704
St. Dev.	0.700919	0.696759	0.707365
Min	-1	-1	-0.99999
Max	1	1	0.99969
Median	0.023406	0.030374	0.099863
Skewness	-0.00889	-0.01755	-0.1349
Kurtosis	1.52585	1.540678	1.498679

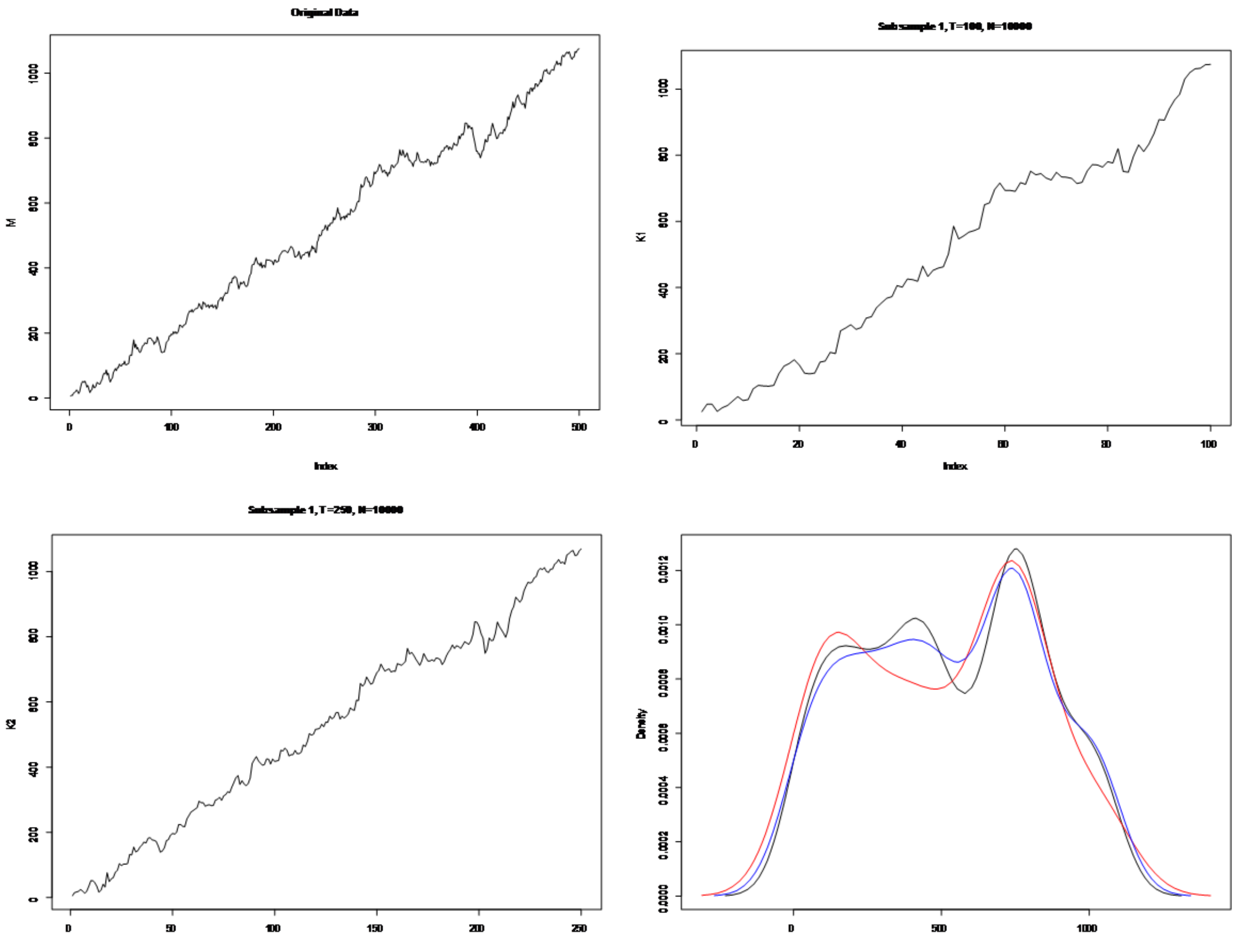
Note: Subsample 1 has 3.685 of the original 10.000 observations, whereas Subsample 2 uses 100.

Source: Authors' calculations based on simulated data

The next step is to investigate how random subsampling works when we have a big data which consists of $\mathrm{T} = 500$ time stamps with $\mathrm{N} = 10,000$ events occurring at each stamp. We can solve this problem by (i) subsampling the time dimension only, and keeping all events for the selected time stamps, or (ii) subsampling the number of events to include in each time stamp. In all cases we use linear aggregation as the conversion function.

Starting with case (i) we see in Figure 10 that using the 20% and 50% of the time stamps still generates a structured time series which approximates well the original data. The second approach we mention above is more puzzling. Selecting a smaller number of events results in a time series with similar characteristics but not the same scale. This is due to the fact that a considerable smaller number of events is aggregated this time; see Figure 11. However, if the experiment does not require the "scaling" of the subsampled data to be accurate, the above procedure could significantly reduce the dimension problem.

Figure 10: Random subsampling in time stamps



Source: Authors' calculations based on simulated data

Ng (2016) applies the above algorithm using Nielsen scanned prices for specific products; examples include pet food and beer, which also exhibit seasonal characteristics. A second type of subsample used in Ng (2016) is the CUR matrix approximation, part of the Leverage Score Sampling. Given a matrix A a CUR matrix approximation consists of three matrices C, U , and R such that C is made from columns of A, R is made from rows of A, and that the product CUR closely approximates A. Usually the CUR is selected to be a $rank - k$ approximation, which means that C contains k columns of A, R contains k rows of A, and U is a $k - by - k$ matrix. There are many possible CUR matrix approximations, and many CUR matrix approximations for a given rank. We could select a column j with probability $p_j = \frac{\|A^{(j)}\|_2^2}{\|A\|_F^2}$ 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 normalisation.

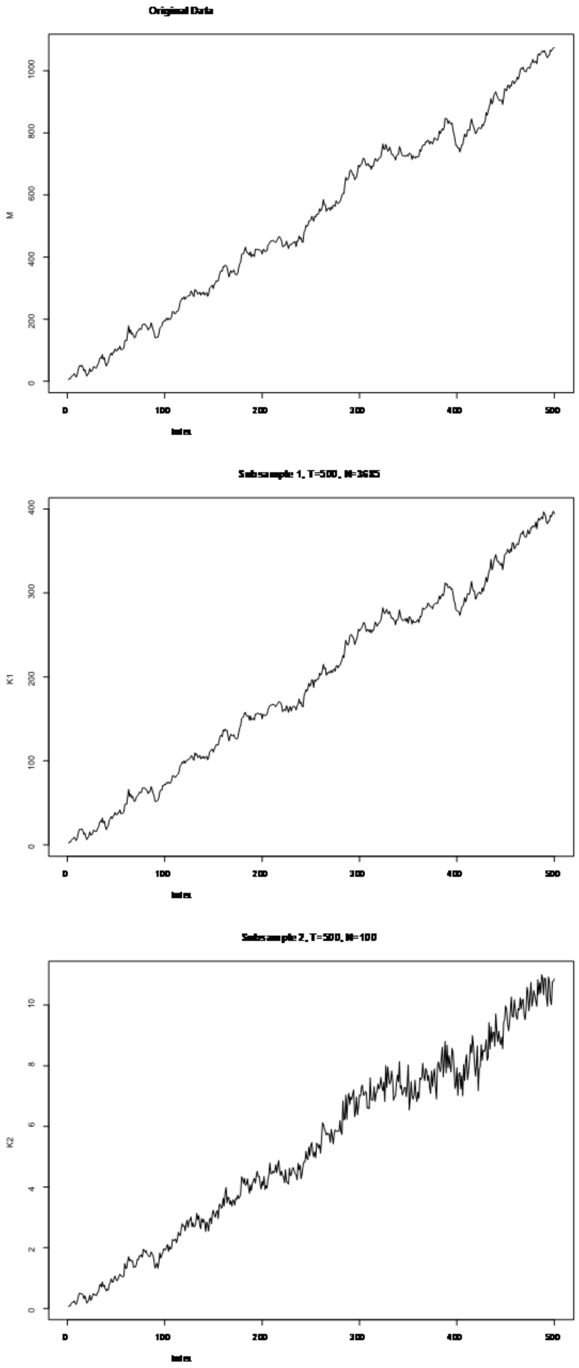
A way to solve this problem is based on the work of Drineas et al. (2008) where we can keep column j with probability $\min(1, \mathrm{c} \cdot \mathrm{pj})$ for some $\mathrm{c} = \mathrm{O}(\mathrm{k} \log \mathrm{k}\epsilon 2)$; also see Mahoney and Drineas (2009). The probability used in this case is given by,

$$p_j = \frac{1}{k}\left\|(V_k^T)^{(j)}\right\|_2^2 \qquad (12)$$

The normalisation by k ensures that pj sums to one. The subspace information Vk are more precise indicators of where information in A is concentrated. Hence when used to define j , they select columns that contain more relevant information about A (leverage score).

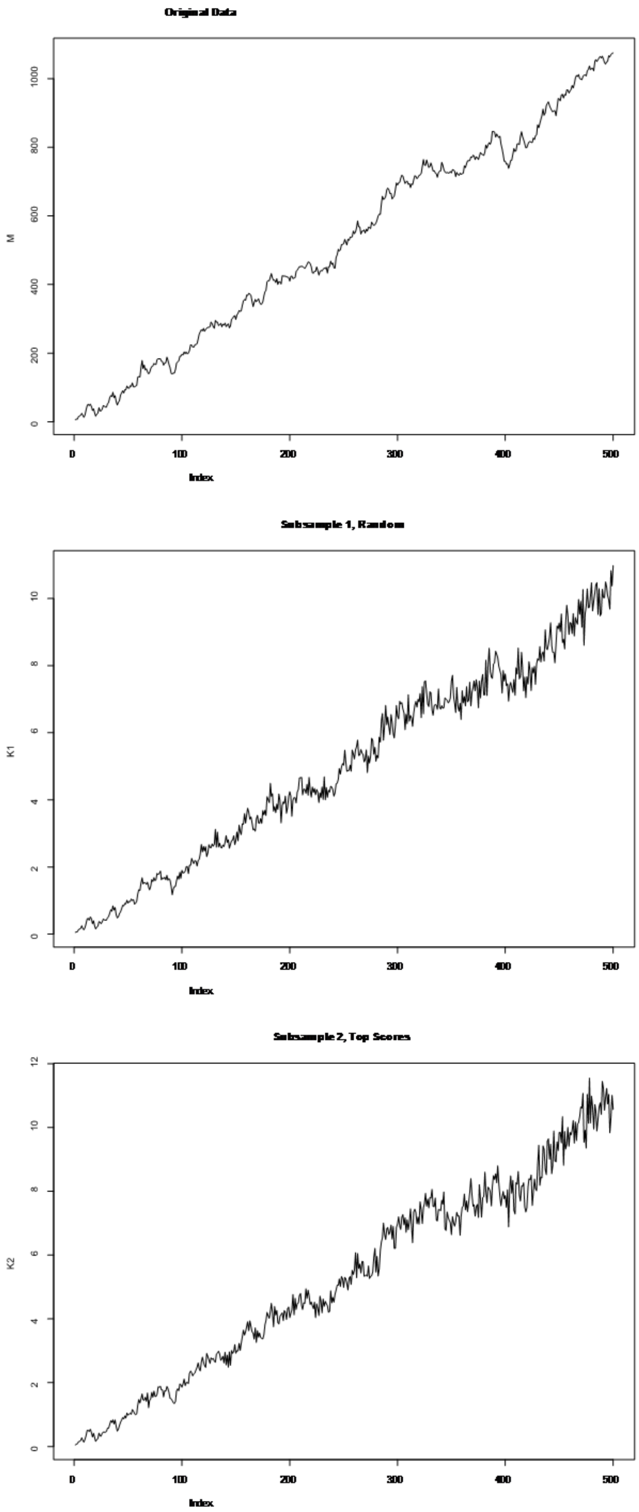
To conclude this section, we continue our example and investigate how well CUR works. Considering the same data as in Figure 11, we use two methods for CUR: (i) random selection of events for all time stamps, (ii) selection of events with top leverage scores. The results are illustrated in Figure 12.

Figure 11: Random subsampling number of events across time



Source: Authors' calculations based on simulated data

Figure 12: CUR subsampling using random and top leverage scores



Source: Authors' calculations based on simulated data

4 Empirical examples

In this section we discuss empirical examples regarding the conversion of unstructured big data to time series format, to complete the theoretical discussion and illustration with simulated data in the previous section.

In each case, we try to provide a conversion example where the resulting time series might be potentially interesting for economic or financial nowcasting.

The examples discussed below are mainly suggested in the literature. Obviously, there can be a lot of different ways each dataset can be converted to time series, but it does not necessarily mean that the output will be suitable for further economic analysis. For example, consider the intraday prices or quotes for a financial instrument. If the purpose of study is the volatility of the security, then time series conversion to realised volatility is a good proxy (more details are discussed in the next subsection). However, if we are interested in the intraday liquidity of the instrument, the frequency of transactions or the aggregate level of transactions might be suitable. Similarly, if the purpose of study is the transaction volume.

4.1 Financial markets data

Realised volatility, commonly used in empirical applications, is based on big datasets of high frequency market quotes or transactions. We use a hypothetical example of tick by tick quotes and trades for an unknown security, labelled XXX, which is being traded simultaneously in ten unknown markets: {B, C, D, I, M , N , P, T, W, X} on the same day. This example mimics NYSE TAQ format, and all steps discussed here can be performed on real data([7]).

At first, a pre-treatment of the raw data is needed. We use the procedure as described in Barndorf-Nielsen, Hansen, Lunde and Shephard (2009). Steps P1-P3 apply to both quotes and trades data types, T1-T4 apply only to trades, whereas Q1-Q4 apply only to quotes.

- All data

  P1. Delete entries with a time stamp outside the 9:30 - 16:00 window when the exchange is open, i.e. exclude the pre-market and after-hours trading (even though, depending on the nature of the experiment a researcher might want to investigate only the pre- or post- market trading sessions).

  P2. Delete data points with a bid, ask or transaction price equal to zero.

  P3. Retain entries originating from a single exchange. Delete other entries or separate them in a similar manner and investigate the different behaviour of the security across multiple exchanges.

- Quotes only

  Q1. When multiple quotes have the same time stamp, replace them with a single entry using the median bid and median ask price.

  Q2. Delete entries for which the bid-ask spread is negative.

([7]) We use data and functions from the “highfrequency" R package: https://goo.gl/S2oJXD

Q3. Delete entries for which the spread is more than 50 times the median spread on that day.

Q4.Delete entries for which the mid-quote deviated by more than 10 mean absolute deviations from a rolling centred median (excluding the observation under consideration) of 50 observations (25 observations before and 25 after).

- Trades only

T1. Delete entries with corrected trades. (Trades with a Correction Indicator, CORR ≠ 0)

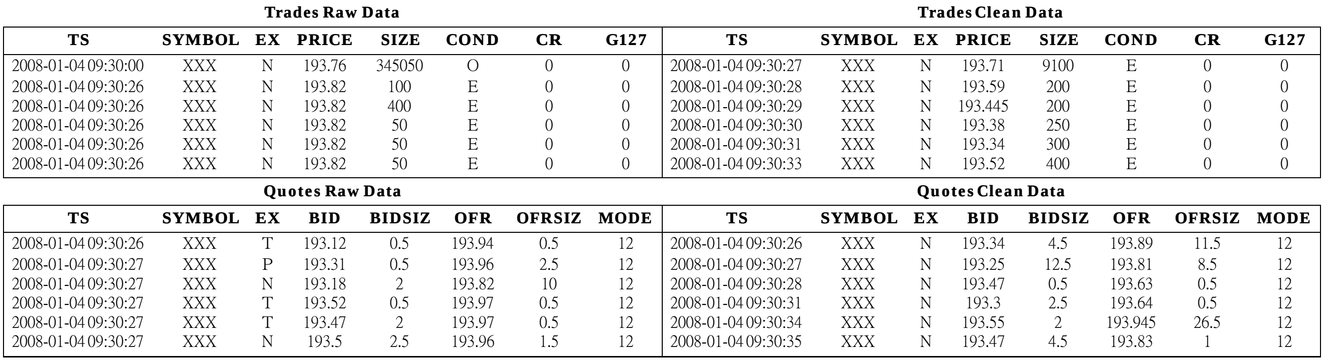
T2. Delete entries with abnormal Sale Condition. (Trades where COND has a letter code, except for E and F).

T3. If multiple transactions have the same time stamp, use the median price.

T4. Delete entries with prices that are above the *ask* plus the *bid-ask* spread. Similar for entries with prices below the *bid* minus the *bid-ask* spread.

A sample of the quotes and prices data is presented in Table 2.

Table 2: Raw and cleaned data for a hypothetical security with ticker XXX



Trades Raw Data

TS	SYMBOL	EX	PRICE	SIZE	COND	CR	G127
2008-01-04 09:30:00	XXX	N	193.76	345050	O	0	0
2008-01-04 09:30:26	XXX	N	193.82	100	E	0	0
2008-01-04 09:30:26	XXX	N	193.82	400	E	0	0
2008-01-04 09:30:26	XXX	N	193.82	50	E	0	0
2008-01-04 09:30:26	XXX	N	193.82	50	E	0	0
2008-01-04 09:30:26	XXX	N	193.82	50	E	0	0

Trades Clean Data

TS	SYMBOL	EX	PRICE	SIZE	COND	CR	G127
2008-01-04 09:30:27	XXX	N	193.71	9100	E	0	0
2008-01-04 09:30:28	XXX	N	193.59	200	E	0	0
2008-01-04 09:30:29	XXX	N	193.445	200	E	0	0
2008-01-04 09:30:30	XXX	N	193.38	250	E	0	0
2008-01-04 09:30:31	XXX	N	193.34	300	E	0	0
2008-01-04 09:30:33	XXX	N	193.52	400	E	0	0

Quotes Raw Data

TS	SYMBOL	EX	BID	BIDSIZ	OFR	OFRSIZ	MODE
2008-01-04 09:30:26	XXX	T	193.12	0.5	193.94	0.5	12
2008-01-04 09:30:27	XXX	P	193.31	0.5	193.96	2.5	12
2008-01-04 09:30:27	XXX	N	193.18	2	193.82	10	12
2008-01-04 09:30:27	XXX	T	193.52	0.5	193.97	0.5	12
2008-01-04 09:30:27	XXX	T	193.47	2	193.97	0.5	12
2008-01-04 09:30:27	XXX	N	193.5	2.5	193.96	1.5	12

Quotes Clean Data

TS	SYMBOL	EX	BID	BIDSIZ	OFR	OFRSIZ	MODE
2008-01-04 09:30:26	XXX	N	193.34	4.5	193.89	11.5	12
2008-01-04 09:30:27	XXX	N	193.25	12.5	193.81	8.5	12
2008-01-04 09:30:28	XXX	N	193.47	0.5	193.63	0.5	12
2008-01-04 09:30:31	XXX	N	193.3	2.5	193.64	0.5	12
2008-01-04 09:30:34	XXX	N	193.55	2	193.945	26.5	12
2008-01-04 09:30:35	XXX	N	193.47	4.5	193.83	1	12

Table notes. TS: time stamp, SYMBOL: security ticker, EX: exchange, PRICE trade price, SIZE: size of trade in lots, COND: Denotes the sale condition associated with a trade, e.g. cash trade, automatic execution, etc., CR: trade correction indicator, G127: Combined ”G”, Rule 127, and stopped stock trade indicator (it only applies to NYSE), BID: bid price, BIDSIZ: size of bid in lots, OFR: ask price, OFRSIZ: size of ask in lots, MODE: quote condition, e.g. closing quote, fast trading, etc. The right panel displays a sample of the raw data, whereas the left panel is concerned with the clean data.

Source: Authors' calculations based on simulated data using "highfrequency" package in R

Let us assume that we are interested in the specific stock exchange *N*. After applying the pre-treatment procedure and obtaining the clean data for the exchange *N*, we end up with 7,706 bid-ask quotes and 8,153 trades for the security XXX. The quotes and trades occur at infrequent points in time.

Looking at Table 2, we see that one trade occurs at 09:30:31 whereas the next one occurs at 09:30:33. Imagine we have a similar dataset for *T* days. How should we convert this unstructured big data into a time series of regularly spaced observations? A commonly used approach underlies the construction of measures of realized volatility (RV), which have several applications in finance but could also be used in a macroeconomic forecasting exercise as indicators of financial uncertainty.

According to Liu, Patton and Sheppard (2015), the most robust measure of realised volatility is the 5-minute RV. The next step is to sample the prices given the preferred frequency. Using the 5-minutes sampling, we are extracting the first price observation in the time stamp which is closest to 09:30:00, 09:35:00, 09:40:00, ..., 15:55:00, 16:00:00. Obviously, we can do the sample for 09:31:00, 09:36:00, 09:41:00, ..., 15:56:00, 16:00:00. However, we stick to the first approach.

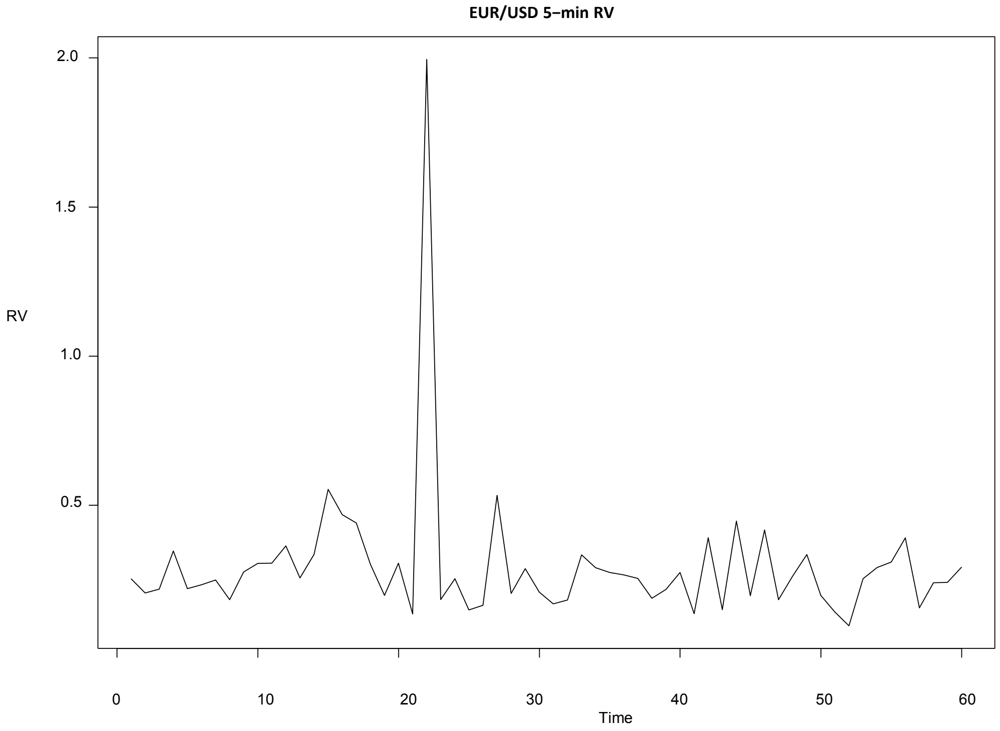
Sampling at 5-minutes provides us with 76 5-min. trade observations on that particular day. The final step is to calculate the 5-min. returns from the 5-min. prices and aggregate their squares. The realised volatility on a particular day is then given by:

$$RV_{5m} = \sum r^2_{j,5m},$$

where r$5m$ denotes the log-return calculated using the 5-min. trade prices. A similar computation can be performed using the mid-price of quotes, and this case the realised volatility will be based on the bid-ask quotes instead of trade prices.

As an illustrative empirical exercise, we report the 5-min realised volatility of the EUR/USD exchange rate for 60 trading days in Figure 13.

Figure 13: Time series of 5-min realised volatility



Source: Authors' calculations based on simulated data using "highfrequency" package in R

4.2 Mobile phone usage data

Mobile phones data is generally not publicly available. For illustrative purposes, we use a sample of mobile phone activity for the city of Milan from Nov. 01, 2013 to Nov. 07, 2013. The data is made publicly available by Telecom Italia as part of the first Big Data Challenge([8]). In total, we have 15,089,165 available observations during the period under consideration, which covers one week. A sample of the data is provided in the following table.

([8]) It can be downloaded at https://goo.gl/rmJfJV

Table 3: Example of mobile phone usage data

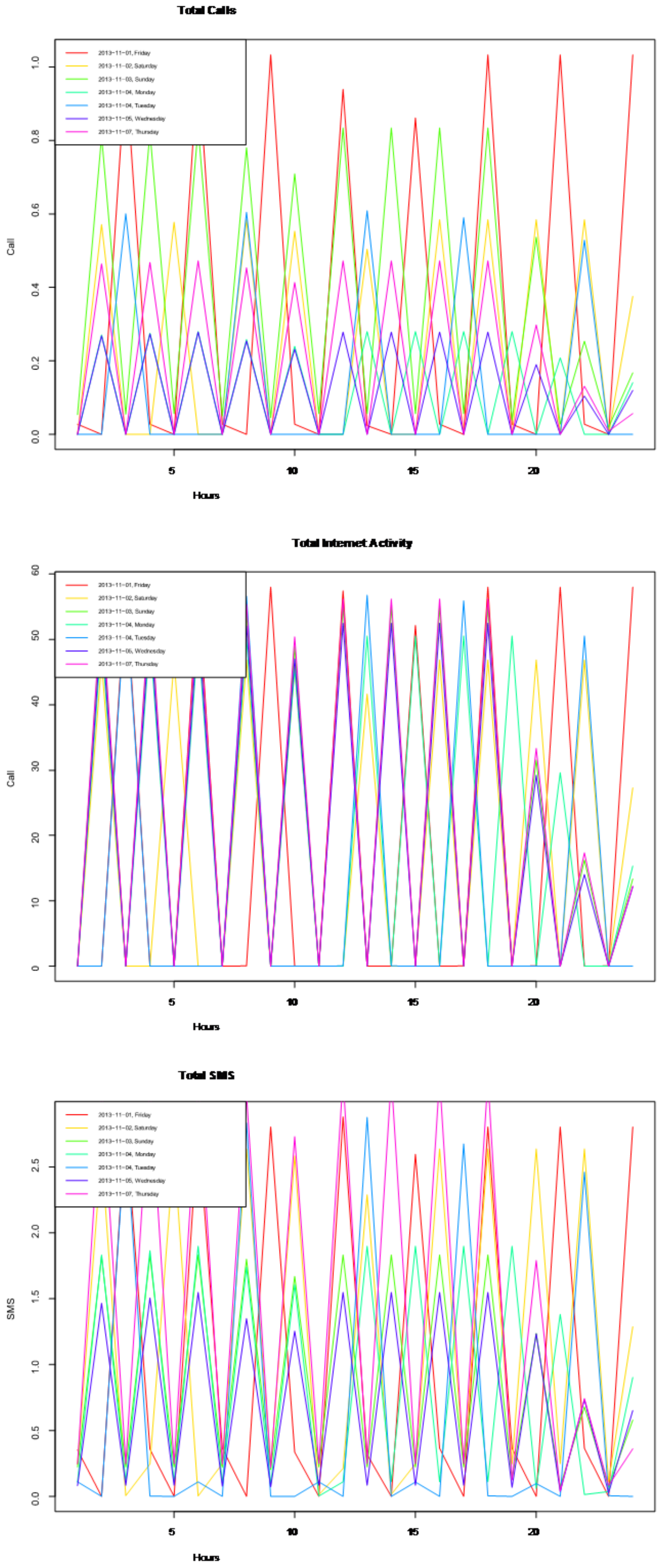
datetime	CellID	Ccode	smsin	smsout	callin	callout	internet
2013-11-07 00:00	1	0	0.2463	NA	NA	NA	NA
2013-11-07 00:00	1	39	1.4627	1.661	0.2999	0.1644	55.8591
2013-11-07 00:00	2	0	0.2467	NA	NA	NA	NA
2013-11-07 00:00	2	39	1.4768	1.6736	0.3033	0.1648	56.0243
2013-11-07 00:00	3	0	0.2471	NA	NA	NA	NA
2013-11-07 00:00	3	39	1.4918	1.687	0.3071	0.1652	56.2002

Note: CellID: the ID of the mobile phone, CCode: the country code, smsin: incoming short text messages, smsout: outgoing short text messages, callin: incoming calls, callout: outgoing calls, internet: mobile internet activity.

Source: Authors' calculations using a sample from Telecom Italia Big Data Challenge 2013

Before transforming the unstructured data to time series format, we have to think about the context that the new time series will be put in. For macroeconomic nowcasting, or nowcasting retail variables, or telecommunications variables, a meaningful approach would be to aggregate the data. First, we can aggregate it across the 24 hours of each day, in order to evaluate the pattern of activity during each day (of the week). This results in 7 hourly time series, reported in the panels of Figure 14.

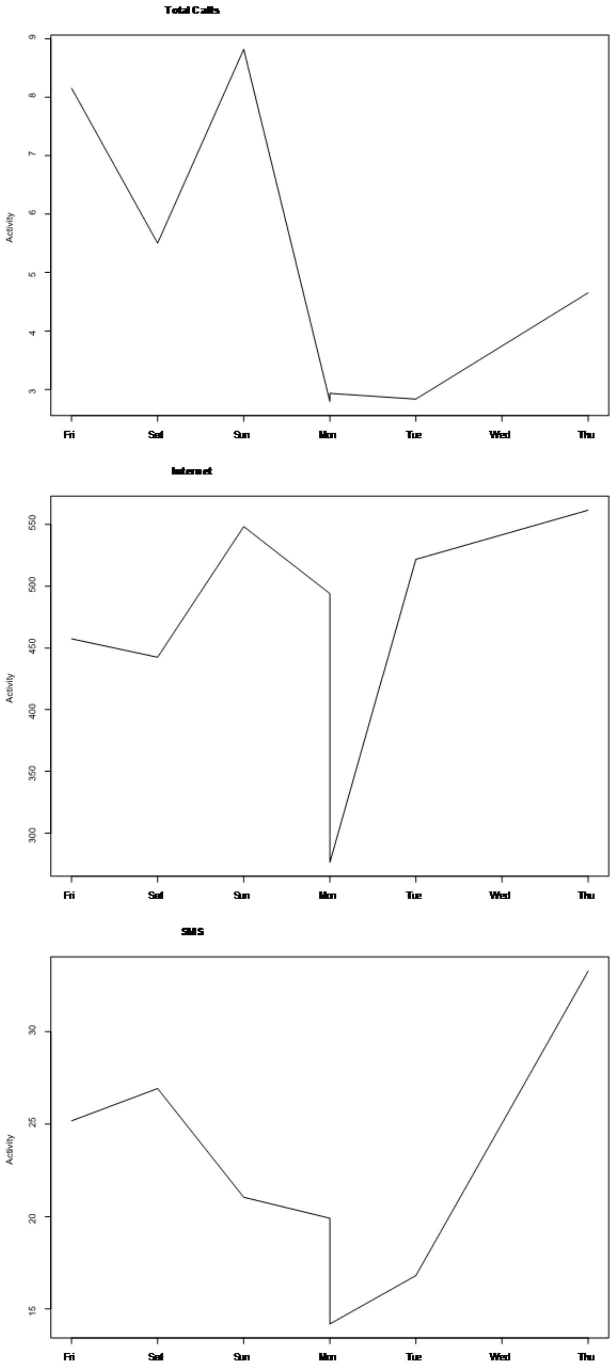
Figure 14: Total mobile activity per hour for different days



Source: Authors' calculations using a sample from Telecom Italia Big Data Challenge 2013

In Figure 14, we observe an intraday periodicity, with increasing and decreasing activity in the total number of calls (incoming and outgoing), internet activity and total SMS (sent and received). Normally, and depending on the nature of the data, we can arrange for a pre-treatment at this point and, for example, either consider the low or zero activity as missing values and interpolate it or specify an econometric model that specifically allows for periods of low activity. As a further alternative, we could proceed with a further aggregation of the data to a daily basis, which gives us a single daily time series set for each mobile phone activity (calls, internet and SMS). These are reported in the panels of Figure 15 for the specific week in our sample.

Figure 15: Total mobile activity per day across time



Source: Authors' calculations using a sample from Telecom Italia Big Data Challenge 2013

From Figure 15, we see that, naturally, the intra-daily temporal pattern is no longer present, but more calls are made during the end of the week, with a large drop on Mondays, in coincidence with the start of the business week. This pattern is more evident in calls and SMS and less evident in internet activity, where Monday seems to be an outlier. We can guess that, if more data were collected, a similar pattern would emerge for other weeks, generating a particular seasonal pattern. After removing or modelling such a pattern, the resulting time series could be used as a coincident indicator for economic activity, either at the city level or perhaps even at the national level, given the importance of Milan for the Italian economy. An even better indicator could be obtained if data on business calls only were available.

4.3 Sensor data

The next empirical example refers to the sensor data type. Most publicly available data for this category includes meteorological measurements, such as temperature, wind direction, wind speed, etc., air traffic control, CO_2 emissions[9], which are not very useful for economic forecasting unless some economic variables depend on weather conditions. To demonstrate the conversion of sensor data to time series structure we use the GeoLife GPS Trajectories[10] as in Zheng, Li, Chen, Xie and Ma (2008), Zheng, Zhang, Xie and Ma (2009), and Zheng, Xie, Ma (2010).

The dataset was collected in Asia (as part of Microsoft Research Asia project) and monitors the GPS location of 182 users in a period of over five years (from April 2007 to August 2012). This dataset recorded a broad range of users' outdoor movements, including not only life routines, like going home or to work, but also some entertainments and sports activities, such as shopping, sightseeing, dining, hiking, and cycling. In the description of the dataset, the provider argues that this dataset can be used in many research fields, such as mobility pattern mining, user activity recognition, location-based social networks, location privacy, and location recommendation. If such a dataset were available for a larger number of individuals, their shopping, dining or similar activities could be used in the nowcasting of retail trade and services. The dataset has the format described in Table 4.

Table 4: Sample data for GPS using mobile phones

Latitude	Longitude	Flag	Altitude (ft)	Date F	Date	Time
39.97440892	116.3035221	0	480.2873556	40753.53069	2011-07-29	12:44:12
39.97439708	116.3035269	0	480.1211516	40753.53071	2011-07-29	12:44:13
39.97398252	116.3036218	0	478.4994554	40753.53073	2011-07-29	12:44:15
39.97394329	116.3036326	0	479.1769882	40753.53074	2011-07-29	12:44:16
39.97393715	116.3036397	0	479.1294324	40753.53075	2011-07-29	12:44:17
39.97391672	116.3036385	0	479.6152789	40753.53076	2011-07-29	12:44:18
39.97389226	116.3036449	0	480.5060269	40753.53078	2011-07-29	12:44:19
39.9738674	116.3036471	0	481.3875098	40753.53079	2011-07-29	12:44:20
39.97383646	116.3036502	0	482.008727	40753.5308	2011-07-29	12:44:21
39.9738212	116.3036494	0	482.3258169	40753.53081	2011-07-29	12:44:22

Notes: Flag: a flag variable always set to zero as in the data guidelines — Date F: fractional format of date — Time is in GMT. The above is just a sample for user No 100.

Source: Authors' calculations based on a sample from GeoLife GPS Trajectories as in Zheng, Li, Chen, Xie and Ma (2008), Zheng, Zhang, Xie and Ma (2009), and Zheng, Xie, Ma (2010)

Table 4 shows the exact location of the user in (almost) every second. In order for a dataset like this to be meaningful for economic nowcasting, we could cross-compare the location of the user to another dataset which contains the latitude, longitude and altitude of known buildings, stores, or other locations of interest. In our example sample, we use 1,976 tracked GPS locations for user 182 during July 29, 2011. It is obvious that taking

[9] Some publicly available data examples can be found online at: https://goo.gl/TVo8bU, https://goo.gl/m0b5gH and https://goo.gl/bBA05U

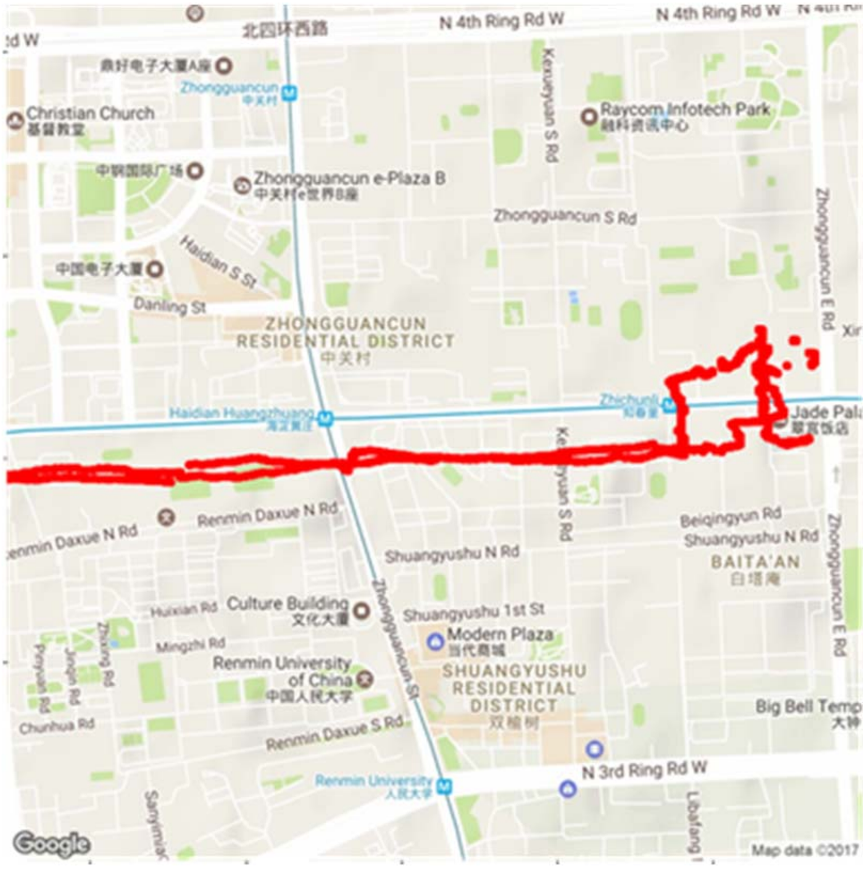
[10] Available at: https://goo.gl/GrXkoB

into account a user's behaviour across a year, it leads us to approximately 430,000 observations or more. Assuming that a representative sample of citizens in a metropolitan area consists of, say, 1,000 citizens, we end up with a sample of 430,000,000 observations per year.

In an effort to reduce the dimension of the data, which needs to be matched with exact locations and, given that GPS sensors in mobile phones are very sensitive, i.e. latitude and longitude change even when humans are not actually “moving", we suggest the use of clustering. Indeed, this is an example where clustering can prove very useful.

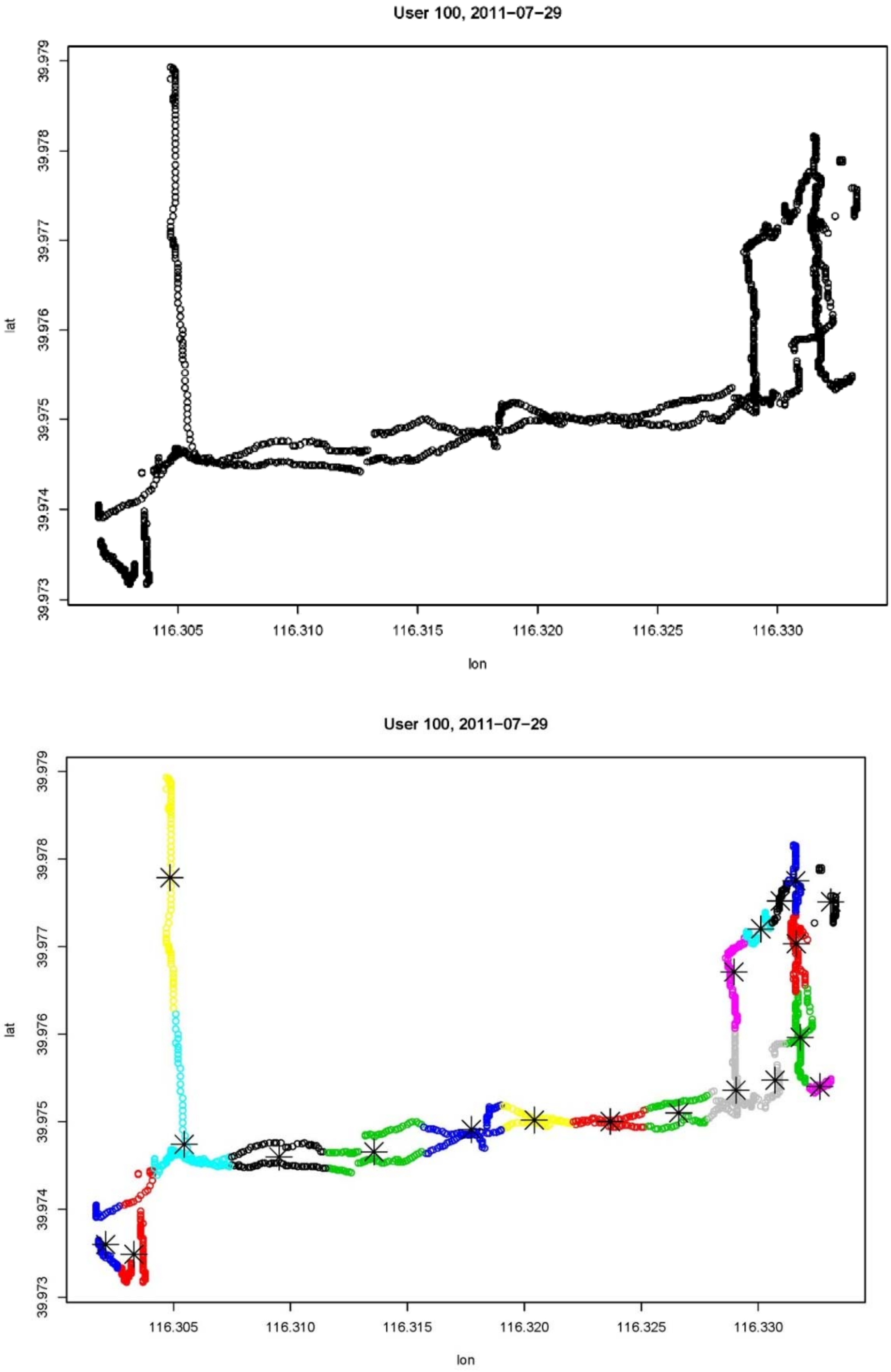
At first, we can plot all the locations that the individual has visited over the specified time interval (one day in our case); see Figure 16. This allows us to track the behaviour of the user and investigate the visited places. A possible way to reduce the dimensionality of the data is to use data mining. As an example, we use the k-clusters mean, presented in the previous section. The elbow method returns three clusters as the optimal number. However, we can increase the number of clusters and get a more meaningful result; see Figure 17 for k-means clustering and Figure 18 for APC.

Figure 16: Tracking an individual



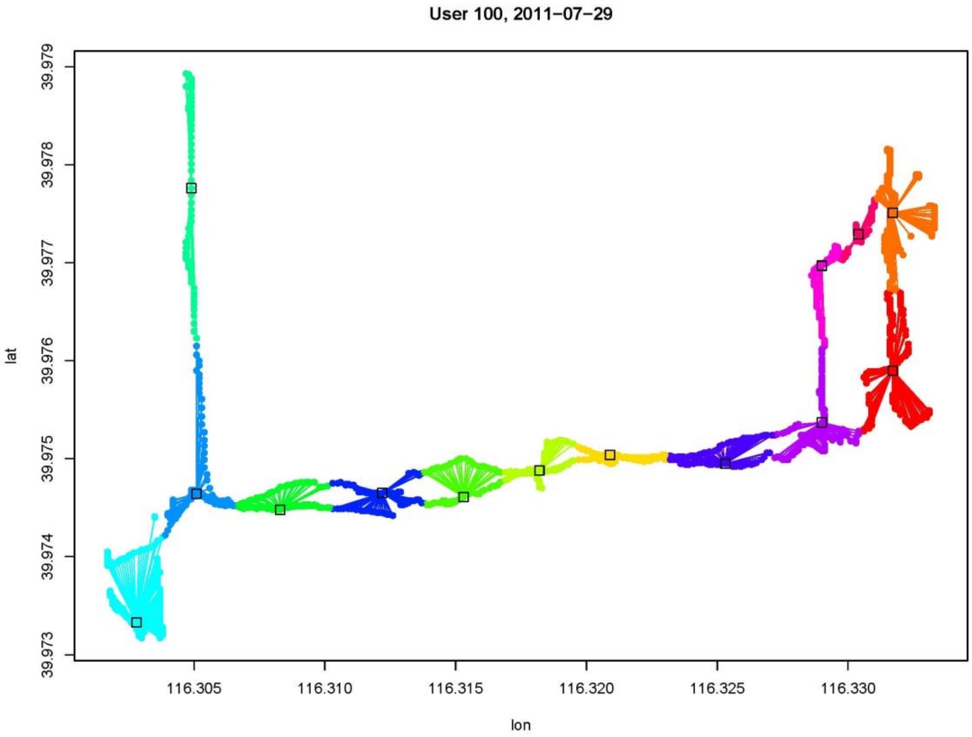
Source: Authors' calculations based on a sample from GeoLife GPS Trajectories as in Zheng, Li, Chen, Xie and Ma (2008), Zheng, Zhang, Xie and Ma (2009), and Zheng, Xie, Ma (2010).

Figure 17: Using k-means clustering to structure the data



Source: Authors' calculations based on a sample from GeoLife GPS Trajectories as in Zheng, Li, Chen, Xie and Ma (2008), Zheng, Zhang, Xie and Ma (2009), and Zheng, Xie, Ma (2010)

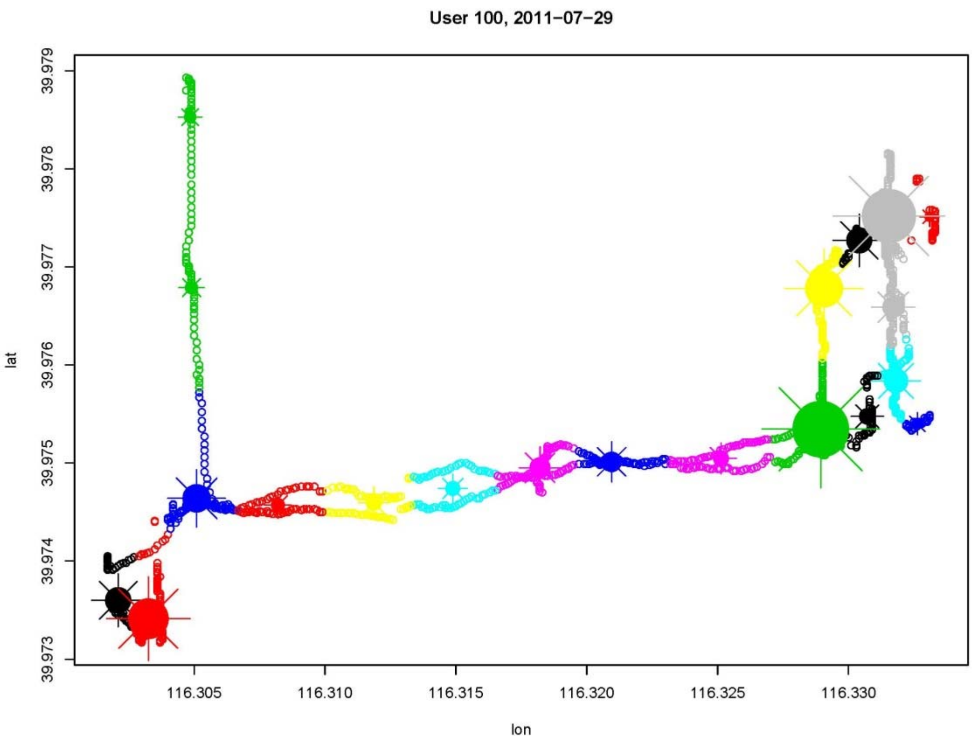
Figure 18: Using APC to structure the data



Source: Authors' calculations based on a sample from GeoLife GPS Trajectories as in Zheng, Li, Chen, Xie and Ma (2008), Zheng, Zhang, Xie and Ma (2009), and Zheng, Xie, Ma (2010)

We can also combine k-means clustering and averaging by allocating a larger weight to clusters associated with locations where the user spent more time. The difference in time spent can be seen in Figure 19.

Figure 19: Using k-means clustering to structure the data



Source: Authors' calculations based on a sample from GeoLife GPS Trajectories as in Zheng, Li, Chen, Xie and Ma (2008), Zheng, Zhang, Xie and Ma (2009), and Zheng, Xie, Ma (2010)

Having a dataset with the location of stores, we could combine it with the above procedure and measure how many times a user visited (or the k-means centre is very close to) a specific shop. Then, we could aggregate across users and/or time to end up with a time series which can potentially improve retail trade nowcasting.

4.4 Online prices data

An interesting approach to collect big data for macroeconomic nowcasting is based on web scraping. Web scraping refers to the automated procedure of scraping information from websites. This can be particularly useful in the nowcasting of inflation, as in Cavallo (2013), one of the main papers based on the output of the Billion Prices Project (BPP)([11]).

In the BPP, prices are scraped on a daily basis from the websites of selected online supermarkets or online retailers (such as Amazon). Therefore, the data is considered as "big", not because of the frequency of occurring observations (as, e.g. in intraday financial prices) but as prices are collected for P products across R retailers resulting in a PxR daily sample, where both P and R are large. For example, a collection of daily prices of, say, 10,000 products from 5 retailers leads to a daily sample of 50,000 observations. On a weekly basis, as in the case of CPI nowcasting, this means about 250,000 observations. We provide more details on web scraping in the next section of this paper.

([11]) See this website for more information: http://bpp.mit.edu/

Table 5: Sample data of web scraped prices

id	day	month	year	date	fullprice	category
1	8	8	2009	40033	2	33
1	9	8	2009	40034	2	33
1	10	8	2009	40035	2	33
1	11	8	2009	40036	2	33
1	12	8	2009	40037	2	33
1	13	8	2009	40038	2	33
1	14	8	2009	40039	2	33
1	15	8	2009	40040	2	33
1	16	8	2009	40041	2	33
1	17	8	2009	40042	2	33

Notes. id: product id — datr: date in fractional format — fullprice: price — category: product's category.

Source: Cavallo (2013)

A sample of the BPP data is presented in Table 5. As in Cavallo (2013), at each time t the unstructured data can be grouped into a price index, obtaining a structured time series which tracks the official CPI. More specifically, the webscraping - online prices based index can be constructed as follows.

1. Let $p_t^{i,j}$ denote the price of good i which belongs to category j at time t. The first step is to calculate the geometric average of price changes across categories:

   $$R_t^j = \prod_i \left(\frac{p_t^{i,j}}{p_{t-1}^{i,j}}\right)^{1/n_t^j}, \qquad (13)$$

   where n_t^j is the number of products in category j available on day t.

2. The next step is to compute the category-level index across time as:

   $$I_t^j = R_1^j . R_2^j \ldots R_t^j \qquad (14)$$

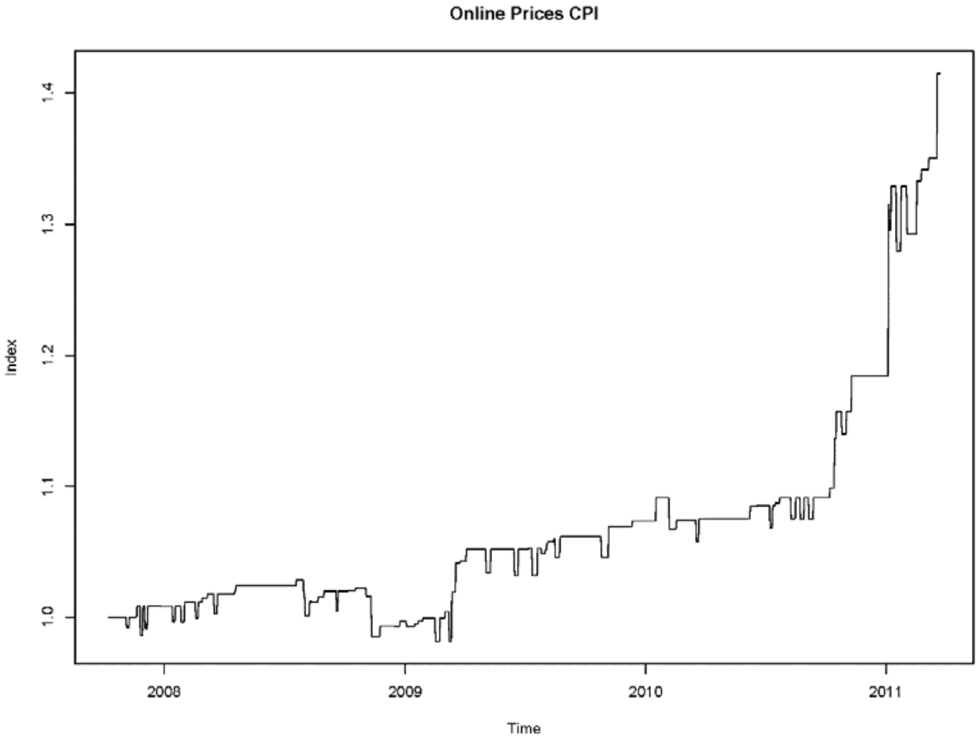
3. Finally, the CPI index based on scraped prices is computed as the weighted arithmetic average across all category indexes:

   $$S_t = \sum_j w^j I_t^j$$

Having the weights across categories, wj , taking values similar to those in the official CPI, makes the online price index directly comparable to the official one. Cavallo (2013) illustrates the correlation between online and offline CPIs, however a proper nowcasting exercise is yet to be done.

The application of the above algorithm to a small sample of four products across three categories yields the structured time series of Figure 20. All data is for Argentina.

Figure 20: CPI index based on web-scraped prices



Source: Authors' calculations based on data from Cavallo (2013)

It must be highlighted that the above discussion also holds in the case of scanner prices data. The only difference is in the data collection. Potentially, CPI nowcasting could benefit from this type of granular, big data based, price indexes.

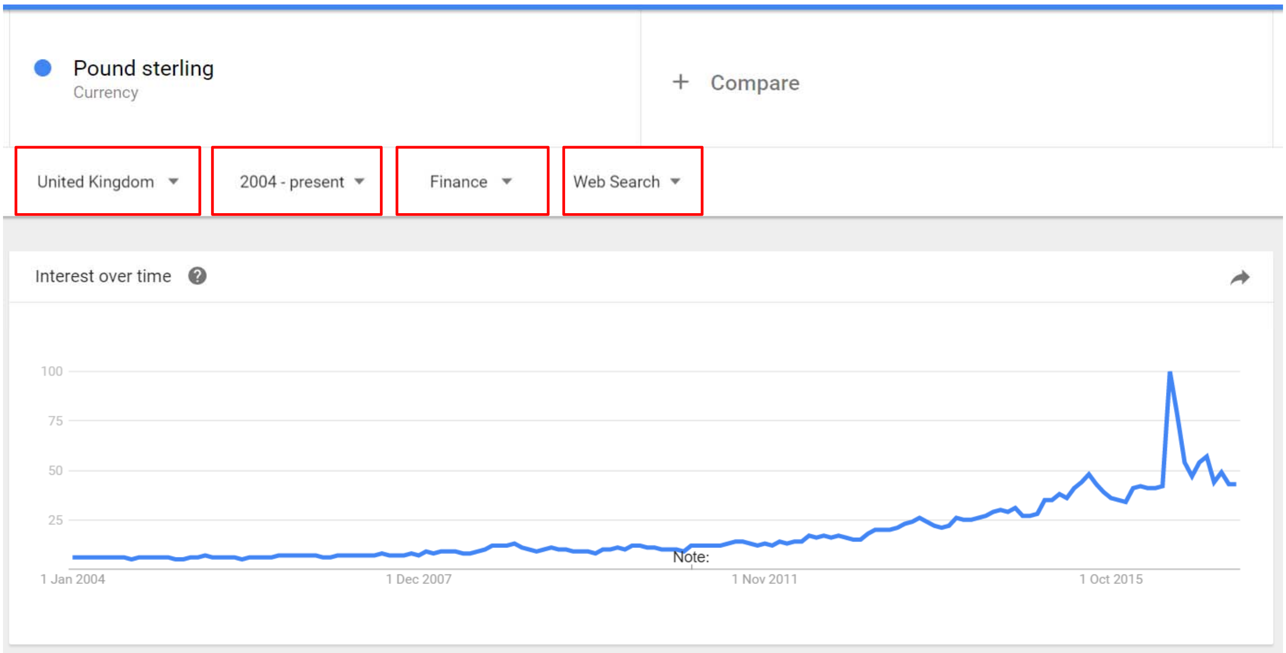
4.5 Online search data

The next big data type we consider is online search data. The prominent source for this data is Google Inc., which has been the most frequently used web search engine during the past twenty years. However, data from other search engines, such as Baidu in China, has also been investigated in the literature with encouraging results. The main purpose of online search data is to identify the number of keyword searches by individual users. Raw data is vast as multiple searches can occur by a single user in a minute. Extending the observations across time and users results into a huge unbalanced dataset.

For privacy reasons, the dataset is not available in raw format. Instead, Google offers a web service, Google Trends, which returns aggregated Google search volumes on a weekly basis. The output provided by Google Trends is a structured time series which measures how often a particular keyword was searched relative to the total search volume. A Google Trends output for a unique keyword indicates the search interest relative to the highest point on the chart for the given region and time. A value of 100 is the peak popularity for the term. A value of 50 means that the term is half as popular. Likewise a score of 0 means the term was less than 1\% as popular as the peak. All subsequent series which are added to the plot are displayed in relative terms to each other.

The user can specify the country, time frame, keyword category and type of search in which the keyword appears. The type of searches include: (i) web search, (ii)\ image search, (iii) news search, (iv) Google shopping, (v)\ Youtube search. In the following figure we show an example using an aggregated output of various terms which are associated with the “Pound sterling" currency.

Figure 21: Google Trends user options



● Pound sterling
Currency

+ Compare

United Kingdom ▾ | 2004 - present ▾ | Finance ▾ | Web Search ▾

Note: The above figure illustrates the highlighted areas for country, time frame, category and type of search.

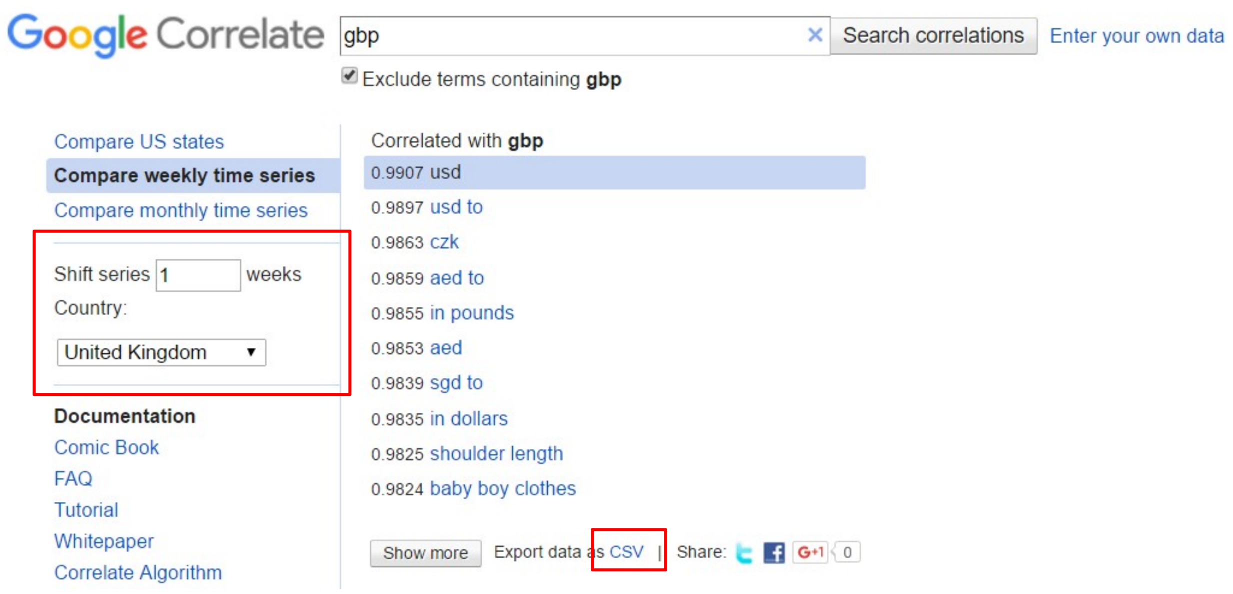
Source: Authors' calculations

The series in Figure 21 can then be extracted in .csv numerical format for further analysis. As it has been noted in various studies, the use of Google search can improve macroeconomic nowcasting([12]). However, the researcher must carefully select the keywords used in the extraction of Google Trends. Also, before using Google Trends in an analysis, their added value should be, conceptually at least, evaluated. For example, as online search data reflects the users' behaviour, it is meaningful to extract keyword searches related to unemployment as most people are looking for job advertisements online. Therefore, an increase in job market related keywords can be associated to the number of people who are currently in search of a new job. Online search data could also contribute to research in the housing sector, holiday destinations and tourism, financial markets (mainly equity prices of specific companies), etc.

If the target consists of more aggregate variables, such as industrial production, retail trade or GDP, then a small number of keywords might not add any value in a nowcasting context. In this case, the researcher might want to consider a large number of keywords, which will result into a big dataset of Google Trend time series. This can be done in, at least, two ways: (i) the first, and safest, is executive action, i.e., the researcher manually creates a universe of keywords and extracts a large number of Google Trends, and (ii) using the web service of Google Correlate; Figure 22 shows the user's options.

([12]) See the discussion in Kapetanios, Marcellino and Papailias (2016)

Figure 22: Google Correlate user options



Google Correlate

gbp × Search correlations Enter your own data

☑ Exclude terms containing **gbp**

Compare US states

Compare weekly time series

Compare monthly time series

Shift series 1 weeks

Country:

United Kingdom ▾

Documentation

Comic Book

FAQ

Tutorial

Whitepaper

Correlate Algorithm

Correlated with **gbp**

0.9907 usd

0.9897 usd to

0.9863 czk

0.9859 aed to

0.9855 in pounds

0.9853 aed

0.9839 sgd to

0.9835 in dollars

0.9825 shoulder length

0.9824 baby boy clothes

Show more Export data as CSV | Share: G+1 0

Source: Authors' calculations

Google Correlate provides a set of keywords of Google Trends which are mostly correlated with the Google Trend of the main keyword of interest, or of a given time series. In the latter case, the researcher can upload the target time series of interest and extract the keywords with the most correlated Google Trends (and of course this can generate a lot of spurious correlation). Continuing our previous "gbp" example for the UK, we see that the Google Trends reported in Table 6 are the most correlated ones.

Table 6: Various correlated keywords to "gbpusd" as returned by Google Correlate

Correlated Google Trends						
gbp	sek to gbp	gbp to huf	a4 b8	off shoulder	reg check	lunch menu
gbp	in dollars	light grey	yen to	js	check if	mm in feet
to gbp	1000 gbp	coconut	aplikacja	tfsi	1d4chan	futa
gbp to usd	shoulder length	cancel	data sim	hair mask	angielsku	with you
usd	baby boy	in ft	inches in cm	pounds in kg	100 dollars	green to
usd to	cake box	huf to gbp	eur	dollars in	celeb news	
usd to gbp	zapain	500 gbp	audi a4 b8	golf gtd	things to draw	
czk	aed to gbp	easy hairstyles	bar menu	best lunch	code 82	
aed to	hkd	yen to gbp	w204	from today	shopify	
in pounds	pounds in	jpy	r53	meetups	amber leaf	
aed	in inches	200 gbp	rest api	feefo	gbp to czk	
czk to gbp	best breakfast	cad to	glute	days from	aud	
pln to gbp	to usd	whose number	dkk to gbp	sgd to gbp	grey and	
gbp to	in cm	300 gbp	unlimited data	for sale gumtree	whose number	
sgd to	angolul	dkk	angel numbers	usd to eur	son quotes	
100 gbp	astra j	3 bedroom	nok to gbp	po angielsku	audi tt mk1	

Source: Authors' calculations based on data from Google

From Table 6, it turns out that this method of keyword selection might yield terms which are irrelevant to the target variable, e.g. "shoulder length", "baby boy clothes", "cake box", "best breakfast". Therefore, human intervention is needed in this step in order to select a criterion to filter the correlated keywords. Then, we could further refine the signal contained only in all the selected keywords.

4.6 Social media

The last empirical example we include in this section of the paper concerns the analysis of textual data. Unstructured textual data can be first transformed to unstructured numerical data, and then the generalised framework described in the previous section can be applied. For example, let us suppose we have a raw unstructured data containing specific keywords searched by a set of internet users. Then, we could specify the universe of keywords which are related to our target variable, select a specific time frame (e.g., every hour), and count the number of times any of the keywords of interest appears in the large textual dataset in the specified time frame. This will result into a structured time series (at hourly frequency).

A recent paper by Gentzkow, Kelly and Taddy (2017) offers some additional insights regarding the use of textual data. Their discussion is in line with the methodologies and results in Kapetanios et al. (2016) and in our previous discussion (in particular, see the subsection on features extraction of numerical data and, more generally, Section 3). As mentioned by Gentzkow, Kelly and Taddy (2017), the first step of the structuring process of textual data is to represent raw text as numerical data. Then, this numerical data is mapped in order to predict values of particular outcomes of interest; in our framework, this is equivalent to the construction of the structured time series based on a feature of the data. Once features are obtained, high-dimensional numerical methods can handle and analyse the resulting dataset. Most of the methods Gentzkow et al. (2017) consider are also discussed in Kapetanios et al. (2016).

We can use the R package “twitteR" to download twitter feeds in R. As an example we use the feed: “gbpusd". The raw output has the following format.

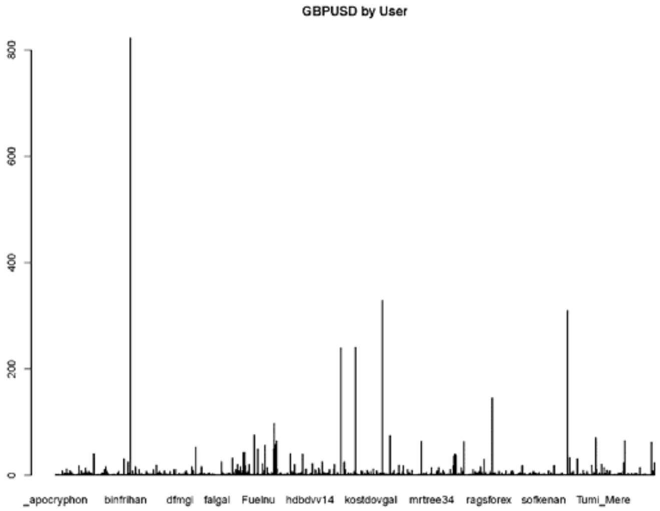
Table 7: Sample Twitter data scraped from Twitter

text	favorited	favoriteCount	replyToSN	created	truncated	ReplyToSID	id	replyToUID	Status Source	screenName	retweetCount	isRetweet	retweeted	longitude	latitude
Donald Tusk	FALSE	0	NA	2017-03-29 13:36:33	FALSE	NA	8.47E+17	NA		TradingintheZen	0	FALSE	FALSE	NA	NA
Forex	FALSE	0	NA	2017-03-29 13:36:01	FALSE	NA	8.47E+17	NA		Loupo85	0	FALSE	FALSE	NA	NA
Forex	FALSE	0	NA	2017-03-29 13:36:00	FALSE	NA	8.47E+17	NA		FranceMob	0	FALSE	FALSE	NA	NA
RT @vpatel2	FALSE	0	NA	2017-03-29 13:33:53	FALSE	NA	8.47E+17	NA		cocuzzo c	4	TRUE	FALSE	NA	NA
Donald Tusk	FALSE	0	NA	2017-03-29 13:30:17	FALSE	NA	8.47E+17	NA		FX14AU	0	FALSE	FALSE	NA	NA
RT @GouldSam	FALSE	0	NA	2017-03-29 13:30:09	FALSE	NA	8.47E+17	NA		idlavi	9	TRUE	FALSE	NA	NA

Source: Authors’ calculations based on data from Twitter

Unfortunately, the current Twitter API returns only the 6,449 most recent tweets. The above data can be turned to structured time series in various ways. If we are interested to perform a user-based analysis, we can count the number of GBPUSD tweets per user across the given time interval; see Figure 23.

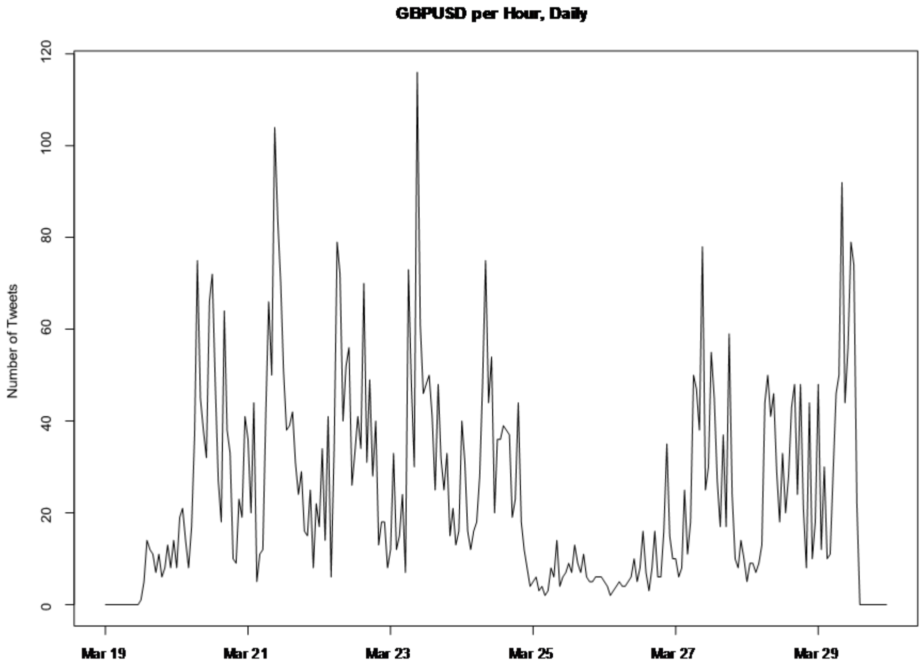
Figure 23: Twitter data by user



Source: Authors’ calculations based on data from Twitter

However, a more interesting approach, better suited for economic and financial analyses, requires the creation of a time series structure, across users, which provides the number of tweets per hour (or other time frames) across each day; see Figure 24.

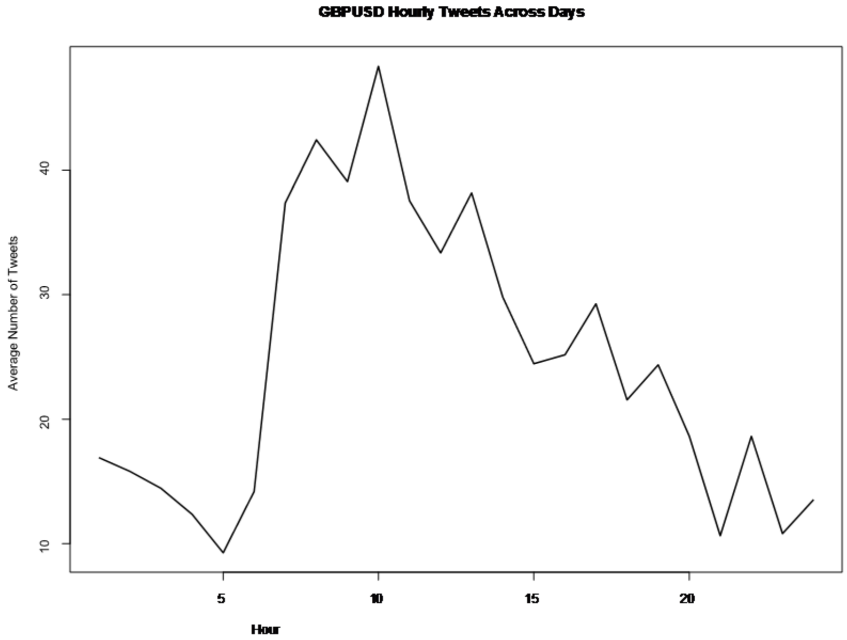
Figure 24: Twitter data per hour across all users for every day



Source: Authors' calculations based on data from Twitter

This results into an hourly time series, which could be used for intraday nowcasting in financial markets. Furthermore, we can construct the average number of GBPUSD tweets per hour across time, which could be the input in a functional analysis to investigate the activity of users during a day; see Figure 25.

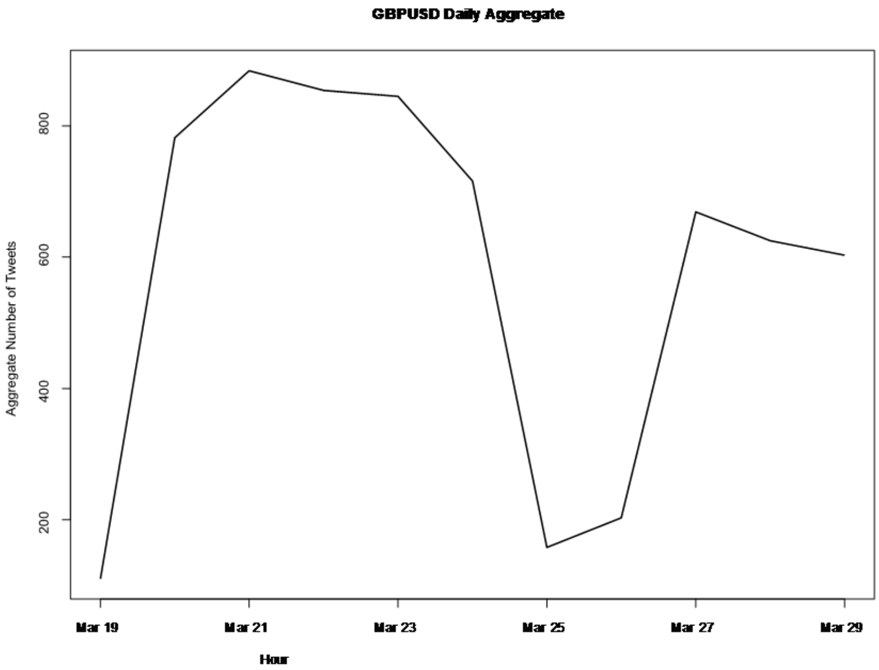
Figure 25: Twitter data by hour across all users and dates



Source: Authors' calculations based on data from Twitter

We could also aggregate all the GBPUSD tweets across hours and obtain a daily time series, which could be also interpreted as a measure of the sentiment of twitter users, which is a concept similar to Google Trends; see Figure 26. This time series can then be directly used in financial and macroeconomic exercises.

Figure 26: Daily time series of Twitter Sentiment



Source: Authors' calculations based on data from Twitter

Finally, as in the case of Google Trends, we can also extract Twitter feeds for related keywords and end up with a large number of Twitter aggregate series, from which a signal can be extracted in a further step.

5 From Reuters News data to uncertainty indexes

After discussing a general approach for structuring big data and showing its implementation on simulated and actual data, in this section we want to further illustrate the structuring process in a more detailed and general application. Specifically, in the context of textual data analysis, we would like to exploit a large set of news articles for the construction of an uncertainty indicator, which can be then used for nowcasting or short-term forecasting economic variables([13]).

Conceptually, we could use Twitter or Google Trends for a similar goal. The main difference is that in Twitter and Google Trends we would analyze users' inputs (tweets and searches respectively), while in this dataset we use published articles. In some sense, it could be argued that Reuters data should be more informative, as it is based on professional journalists and not individuals users (who might or might not be professionals) and it is accessible to a selected but large number of key economic agents, those with access to Reuters terminals.([14])

Our starting big dataset consists of roughly 3 million Reuters News articles, published between 2007 and 2016. The work draws from the growing economic text mining literature and, in particular, from Baker et al. (2016), and Eckley (2016). In the following subsections we describe in detail how a time series measuring macroeconomic uncertainty can be extracted from this vast amount of unstructured textual data. First, we address common challenges in the collection of news data: downloading and storing in a machine-readable format web pages containing texts and removing duplicate articles. While the implementation of our data collection is specific to the Reuters News Archive, the web scraping techniques and the duplicates removal strategy can be applied to all automated analyses of big news data. Then, we show how to obtain a daily index based on frequencies of articles containing uncertainty terms. We also explain critical choices in the selection of words to be included in the search dictionaries. Finally, we briefly describe the obtained structured time series.

5.1 Obtaining the data

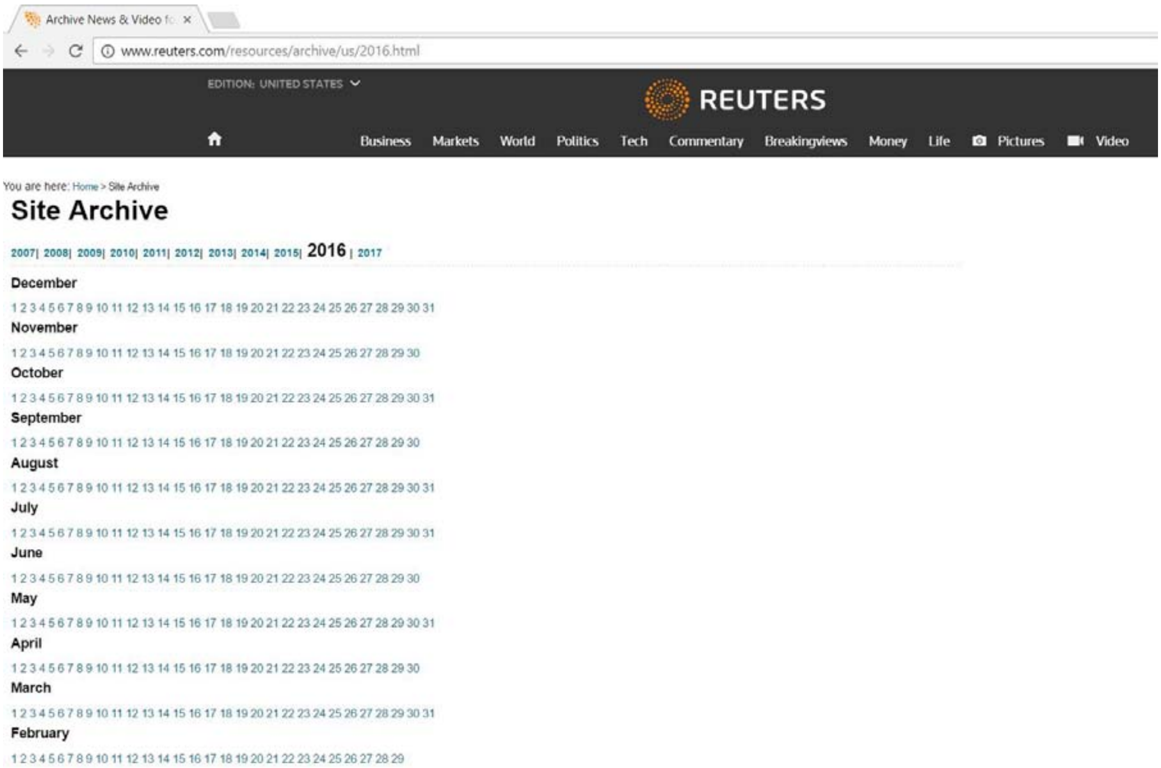
The source for the text data is the Reuters Online News Archive([15]), a collection of news articles published in the United States edition of the Reuters website and freely available on the Internet. The archive coverage starts from January 1st, 2007 and it is updated continuously as content is published on the website. The online repository is structured as a plain HTML web-page, and can be browsed by date (see Figure 27): subpages simply list articles published on a single day in inverse chronological order.

[13] We would like to thank Mattia Serrano for valuable assistance in the preparation of this section.

[14] An obvious way to combine alternative sources would be to select a universe of Twitter users who are proved to have particular knowledge about economic news, e.g. Financial Times journalists, Bloomberg journalists, investors, spokespersons, etc. Then, it is reasonable to argue that their tweets also reflect the views of the articles they are about to write in the near future. This actually could be even more timely for nowcasting purposes.

[15] Please see: http://www.reuters.com/resources/archive/us/

Figure 27: The home page of the Reuters News Archive



You are here: Home > Site Archive

Site Archive

2007| 2008| 2009| 2010| 2011| 2012| 2013| 2014| 2015| 2016 | 2017

December

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

November

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

October

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

September

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

August

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

July

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

June

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

May

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

April

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

March

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

February

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29

Source: Authors' calculations based on data from Reuters

Although the archive is not searchable, its simple design allows for easy web-scraping. However, since articles are listed as they are published and updated, the archive contains duplicates and a large number of links return a 404 error (page not found). We have developed a three-stage web-scraping strategy (presented in the following paragraphs) that addresses these issues by reducing the amount of pages to be downloaded, thus damping noise in the data and minimising the time needed to collect the articles.

The following subsections describe the various stages of the data collection process: listing the news articles (Section 5.2 Obtaining a list of links), duplicates removal (Section 5.3) and scraping of the texts (Section 5.4). The entire data collection is implemented using code written in the Python programming language, version 3.5 for the de-duplication and the data cleaning and version 2.7 for the web-scraping. While the algorithms are designed to work specifically with the Reuters News Archive, the basic working principles of the code apply for most news text sources.

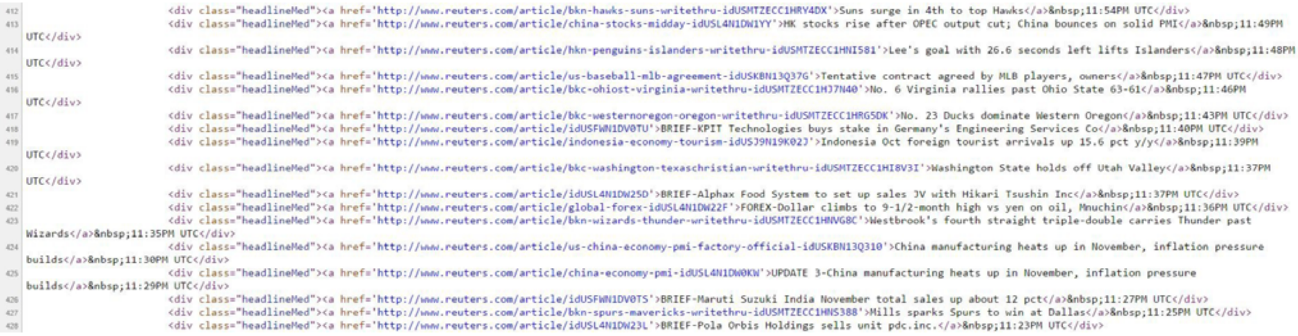
5.2 Obtaining a list of links

The first step of the data collection entails obtaining a list of all the articles stored in the archive. Specifically, our code lists the date, URL (Uniform Resource Locator) and headline of each article as they appear in the archive. This step allows us to remove ex-ante links to articles that are broken or duplicated, without the need to download the article text. Given the large number of duplicates and broken links, this translates into a significantly faster data collection and reduction in memory storage space.

The Python code we implement is a typical web-scraping "spider": it starts from the archive home page and navigates through the different levels of subpages: years, at the first level, and days, at the second. The spider enters a year parent page first, and then it proceeds opening each day subpage one-by-one. It does so by parsing the HTML code of the page and matching a regular expression on URLs that end with a series of numbers representing a date, like http://www.reuters.com/resources/archive/us/20160116.html.

As shown in Figure 28, each day subpage contains the list of all the articles published on that day, starting from the last. At this point, the spider parses the HTML code of the page in order to identify the components we are interested into: the URL and the headline.

Figure 28: The Reuters Archive page for 1 December 2016



```
                    <div class="headlineMed"><a href='http://www.reuters.com/article/bkn-hawks-suns-writethru-idUSMTZECC1HRY4DX'>Suns surge in 4th to top Hawks</a>&nbsp;11:54PM UTC</div>
                    <div class="headlineMed"><a href='http://www.reuters.com/article/china-stocks-midday-idUSL4N1DW1YY'>HK stocks rise after OPEC output cut; China bounces on solid PMI</a>&nbsp;11:49PM
UTC</div>
                    <div class="headlineMed"><a href='http://www.reuters.com/article/hkn-penguins-islanders-writethru-idUSMTZECC1HNI581'>Lee's goal with 26.6 seconds left lifts Islanders</a>&nbsp;11:48PM
UTC</div>
                    <div class="headlineMed"><a href='http://www.reuters.com/article/us-baseball-mlb-agreement-idUSKBN13Q37G'>Tentative contract agreed by MLB players, owners</a>&nbsp;11:47PM UTC</div>
                    <div class="headlineMed"><a href='http://www.reuters.com/article/bkc-ohiost-virginia-writethru-idUSMTZECC1HJ7N40'>No. 6 Virginia rallies past Ohio State 63-61</a>&nbsp;11:46PM
UTC</div>
                    <div class="headlineMed"><a href='http://www.reuters.com/article/bkc-westernoregon-oregon-writethru-idUSMTZECC1HRG5DK'>No. 23 Ducks dominate Western Oregon</a>&nbsp;11:43PM UTC</div>
                    <div class="headlineMed"><a href='http://www.reuters.com/article/idUSFWN1DV0TU'>BRIEF-KPIT Technologies buys stake in Germany's Engineering Services Co</a>&nbsp;11:40PM UTC</div>
                    <div class="headlineMed"><a href='http://www.reuters.com/article/indonesia-economy-tourism-idUSJ9N19K02J'>Indonesia Oct foreign tourist arrivals up 15.6 pct y/y</a>&nbsp;11:39PM
UTC</div>
                    <div class="headlineMed"><a href='http://www.reuters.com/article/bkc-washington-texaschristian-writethru-idUSMTZECC1HI8V3I'>Washington State holds off Utah Valley</a>&nbsp;11:37PM
UTC</div>
                    <div class="headlineMed"><a href='http://www.reuters.com/article/idUSL4N1DW25D'>BRIEF-Alphax Food System to set up sales JV with Hikari Tsushin Inc</a>&nbsp;11:37PM UTC</div>
                    <div class="headlineMed"><a href='http://www.reuters.com/article/global-forex-idUSL4N1DW22F'>FOREX-Dollar climbs to 9-1/2-month high vs yen on oil, Mnuchin</a>&nbsp;11:36PM UTC</div>
                    <div class="headlineMed"><a href='http://www.reuters.com/article/bkn-wizards-thunder-writethru-idUSMTZECC1HNVG8C'>Westbrook's fourth straight triple-double carries Thunder past
Wizards</a>&nbsp;11:35PM UTC</div>
                    <div class="headlineMed"><a href='http://www.reuters.com/article/us-china-economy-pmi-factory-official-idUSKBN13Q310'>China manufacturing heats up in November, inflation pressure
builds</a>&nbsp;11:30PM UTC</div>
                    <div class="headlineMed"><a href='http://www.reuters.com/article/china-economy-pmi-idUSL4N1DW0KW'>UPDATE 3-China manufacturing heats up in November, inflation pressure
builds</a>&nbsp;11:29PM UTC</div>
                    <div class="headlineMed"><a href='http://www.reuters.com/article/idUSFWN1DV0TS'>BRIEF-Maruti Suzuki India November total sales up about 12 pct</a>&nbsp;11:27PM UTC</div>
                    <div class="headlineMed"><a href='http://www.reuters.com/article/bkn-spurs-mavericks-writethru-idUSMTZECC1HNS388'>Mills sparks Spurs to win at Dallas</a>&nbsp;11:25PM UTC</div>
                    <div class="headlineMed"><a href='http://www.reuters.com/article/idUSL4N1DW23L'>BRIEF-Pola Orbis Holdings sells unit pdc.inc.</a>&nbsp;11:23PM UTC</div>
```

Source: Authors' calculations based on data from Reuters

Finally, the date, URL and headline of each article are written to separate lines of a .CSV (comma separated values) file that is used in the subsequent steps.

5.3 Removing duplicates

As observed by Eckley (2016), duplication of articles is a common problem in the computational analysis of data. Examining the Factiva Financial Times records, Eckley finds duplication to be substantial and emerging from a variety of causes, with no single pattern over time or across articles.

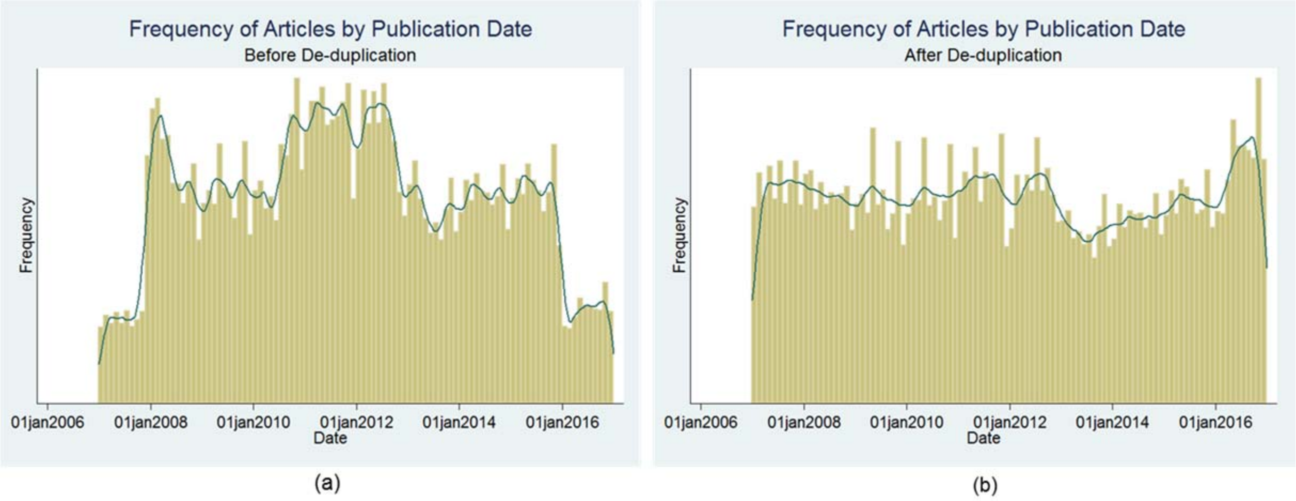
This appears to be the case also in the Reuters News Archive, with an estimated 13.6% of duplicated articles after the removal of links not working (see Table 8). The distribution of the daily count of articles in the dataset, pictured in the left panel of Figure 29, also hints at a significant presence of duplicates. While we would expect the number of articles published on each day to be relatively stable over time, the distribution is very far from uniform. The right panel of Figure 29, on the other hand, shows the same distribution becoming much closer to uniform after de-duplication. The presence of duplicates can introduce biases in the computation of indices based on article counts, as well as increase the level of noise and the time needed to scrape the article texts.

Table 8: Summary of de-duplication of articles

Reuters US News Archive 2007-2016		
Total Articles	8.557.866	
Links not working	5.181.771	60.55%
Working links	**3.376.095**	
Duplicate URLs	2.288	0.07%
Duplicate Headlines	277.651	8.22%
Uppercase Duplicate Headlines	58.962	1.75%
Article Updates	120.963	3.58%
De-duplicated Articles	**2.916.231**	**13.62%**

Source: Authors' calculations based on data from Reuters

Figure 29: The distribution of articles, before and after de-duplication



Source: Authors' calculations based on data from Reuters

Similarly to Eckley (*ibidem*), our de-duplication strategy targets identical duplicates as well as minor or cosmetic variations in articles. Differently from Eckley, however, we attempt to remove duplicate articles before they are downloaded, by comparing article headlines, URLs and publication dates, as obtained above (see Section 5.2). The purpose is twofold: (i) reducing the number of articles to download and (ii) making the de-duplication algorithm leaner by only using criteria based on URLs and headline similarity. We develop our de-duplication algorithm in successive stages, checking manually for false positives (i.e. unique articles identified as duplicates) and false negatives (residual duplicates) after each iteration of the algorithm, with the aim of avoiding the former, while keeping the latter to a minimum.

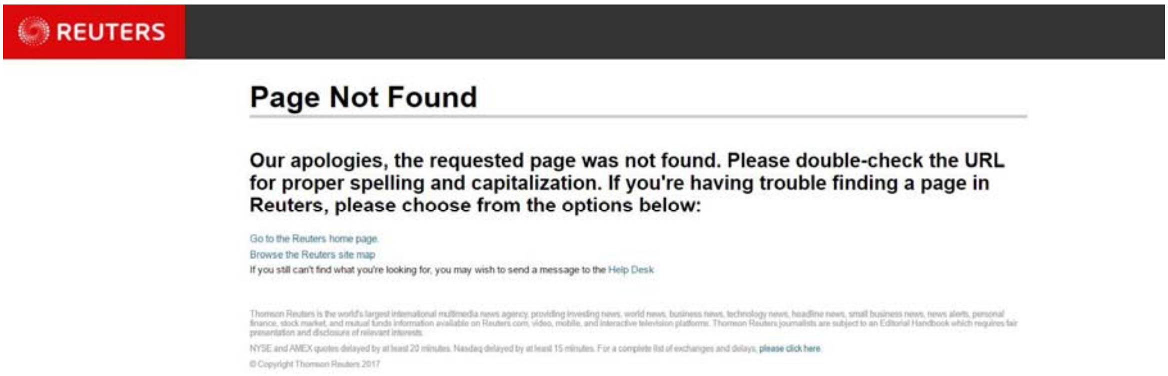
Before delving further in the removal of duplicates, we must first address an issue specific to our dataset: the presence of broken links.

5.3.1 Removing broken links

A broken or dead link is a hyperlink pointing to a webpage that has been deleted or moved. The Reuters Online News Archive contains a very high number of dead links, slightly above 60% of all links, according to our estimates.

Attempting to scrape the text from these links would significantly and unnecessarily slow down the web scraper. Therefore, it would be desirable to remove as many broken links as possible before the actual scraping of the article texts. A manual exploration of the links scraped in Section 5.2 reveals that most of the URLs that return a 404 error (see Figure 30) share some common characteristics that allow them to be distinguished from working links. Specifically, broken links appear to be press releases distributed by major news wire services and re-published by Reuters without any changes. The URL contains the string “PressRelease", a date and a code corresponding to the service that originated the press release, as well as one or more “+" (plus) signs. The latter characteristic, in particular, appears to be unique to the broken links, allowing for a parsimonious and yet effective algorithm that identifies and removes URLs containing plus signs.

Figure 30: A “page not found error” displayed in the Reuters website



Page Not Found

Our apologies, the requested page was not found. Please double-check the URL for proper spelling and capitalization. If you're having trouble finding a page in Reuters, please choose from the options below:

Go to the Reuters home page.
Browse the Reuters site map
If you still can't find what you're looking for, you may wish to send a message to the Help Desk

Thomson Reuters is the world's largest international multimedia news agency, providing investing news, world news, business news, technology news, headline news, small business news, news alerts, personal finance, stock market, and mutual funds information available on Reuters.com, video, mobile, and interactive television platforms. Thomson Reuters journalists are subject to an Editorial Handbook which requires fair presentation and disclosure of relevant interests.

NYSE and AMEX quotes delayed by at least 20 minutes. Nasdaq delayed by at least 15 minutes. For a complete list of exchanges and delays, please click here.

© Copyright Thomson Reuters 2017

Source: Authors’ calculations based on data from Reuters

5.3.2 Removing duplicate URLs

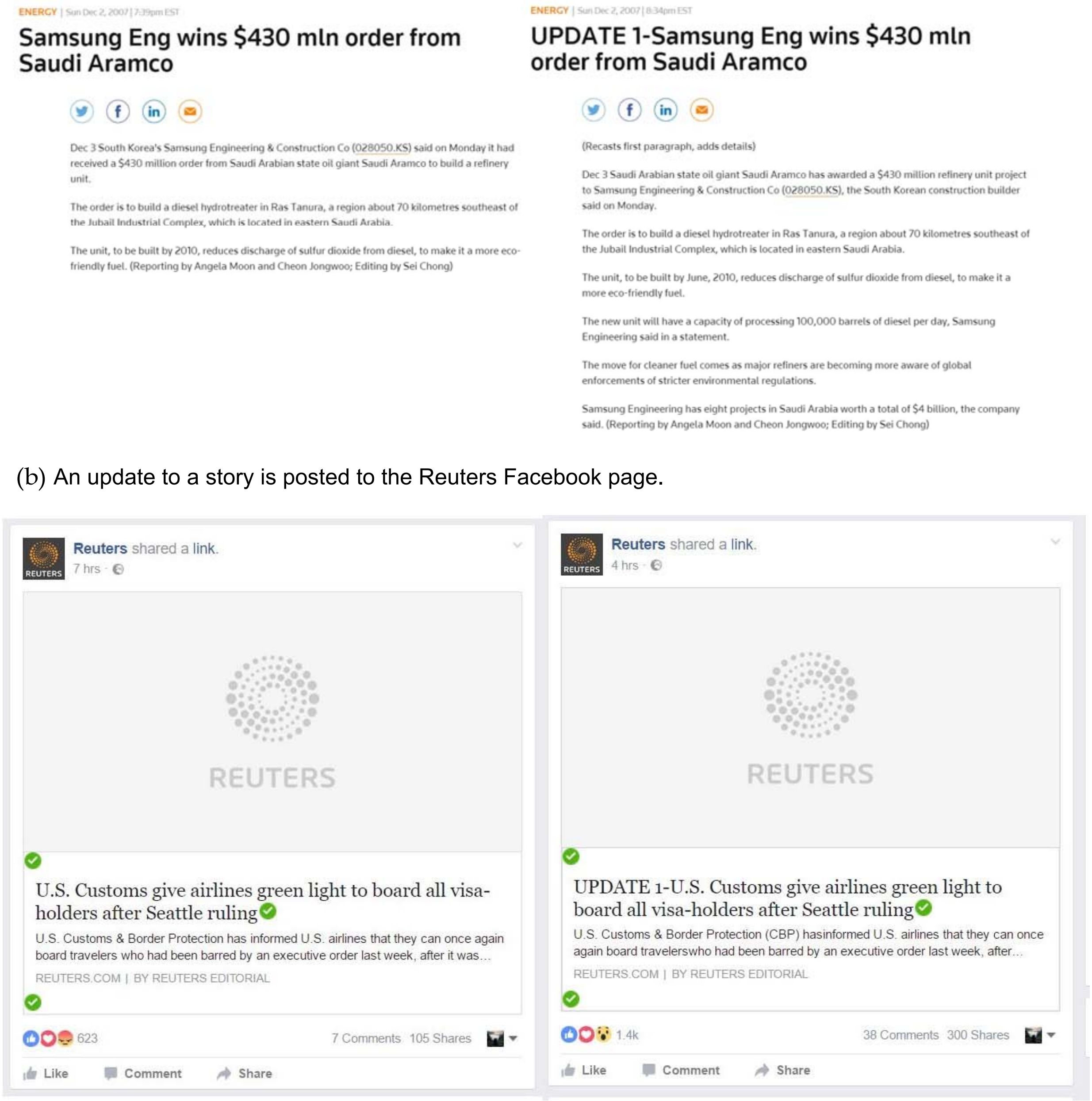
Moving onto proper de-duplication, a simple first step consists in checking for the presence of identical URLs in the archive. As URLs identify uniquely a web-page, if the same URL appears twice in the archive, that is undoubtedly a duplicate. Our algorithm finds 2,288 repeated URLs, mainly concentrated in two days of 2015 and 2014. This appears to be a random error in the archiving of the articles for those days.

5.3.3 Removing duplicate headlines

Following Eckley (2016), we proceed by removing records with identical publication date and headline. The algorithm finds a very large number of repeated headlines (277,651, or about 8% of total working links, see Table 8): each day of news archived contains several such duplicates. Manual examination of a sample of these links reveals that repetitions of this kind usually occur within a few minutes since the first article is published, as news are reposted to different sections of the website or pictures are added to the story. For headlines duplicated in this way, the article body is identical in all instances we examined manually and, therefore, the computation of the index benefits from the deletion.

Although the article bodies are not identical, leaving updated stories in the corpus would bias our index in a way very similar to the duplicates removed through the procedures described above. In fact, an index that counts the fraction of articles expressing uncertainty over the total number of articles published in a given period would count each update as a separate story, whereas it is just the same article being expanded. On the other hand, it would be a mistake to treat updates as any other duplicate, since each additional recast of the story contains more information than before. For example, an article that first contained no reference to economic uncertainty, might be updated to reflect it in successive versions. Therefore, the way updated articles are removed can directly influence the computation of the uncertainty index. Our final headline-parsing algorithm (see provided code) only keeps the last update of a story on a given day, discarding from the corpus the original article and its previous updates. Notice that if a further update of the story is published on the following day, that article is kept in the dataset as it amounts to a new story for that day.

Figure 31: Examples of article updates on the Reuters website and on social media



(a) An article is updated with additional information about one hour after the initial publication.

ENERGY | Sun Dec 2, 2007 | 7:39pm EST

Samsung Eng wins $430 mln order from Saudi Aramco

Dec 3 South Korea's Samsung Engineering & Construction Co (028050.KS) said on Monday it had received a $430 million order from Saudi Arabian state oil giant Saudi Aramco to build a refinery unit.

The order is to build a diesel hydrotreater in Ras Tanura, a region about 70 kilometres southeast of the Jubail Industrial Complex, which is located in eastern Saudi Arabia.

The unit, to be built by 2010, reduces discharge of sulfur dioxide from diesel, to make it a more eco-friendly fuel. (Reporting by Angela Moon and Cheon Jongwoo; Editing by Sei Chong)

ENERGY | Sun Dec 2, 2007 | 8:34pm EST

UPDATE 1-Samsung Eng wins $430 mln order from Saudi Aramco

(Recasts first paragraph, adds details)

Dec 3 Saudi Arabian state oil giant Saudi Aramco has awarded a $430 million refinery unit project to Samsung Engineering & Construction Co (028050.KS), the South Korean construction builder said on Monday.

The order is to build a diesel hydrotreater in Ras Tanura, a region about 70 kilometres southeast of the Jubail Industrial Complex, which is located in eastern Saudi Arabia.

The unit, to be built by June, 2010, reduces discharge of sulfur dioxide from diesel, to make it a more eco-friendly fuel.

The new unit will have a capacity of processing 100,000 barrels of diesel per day, Samsung Engineering said in a statement.

The move for cleaner fuel comes as major refiners are becoming more aware of global enforcements of stricter environmental regulations.

Samsung Engineering has eight projects in Saudi Arabia worth a total of $4 billion, the company said. (Reporting by Angela Moon and Cheon Jongwoo; Editing by Sei Chong)

(b) An update to a story is posted to the Reuters Facebook page.

Reuters shared a link.
7 hrs

U.S. Customs give airlines green light to board all visa-holders after Seattle ruling

U.S. Customs & Border Protection has informed U.S. airlines that they can once again board travelers who had been barred by an executive order last week, after it was…

REUTERS.COM | BY REUTERS EDITORIAL

623 · 7 Comments · 105 Shares

Like · Comment · Share

Reuters shared a link.
4 hrs

UPDATE 1-U.S. Customs give airlines green light to board all visa-holders after Seattle ruling

U.S. Customs & Border Protection (CBP) hasinformed U.S. airlines that they can once again board travelerswho had been barred by an executive order last week, after…

REUTERS.COM | BY REUTERS EDITORIAL

1.4k · 38 Comments · 300 Shares

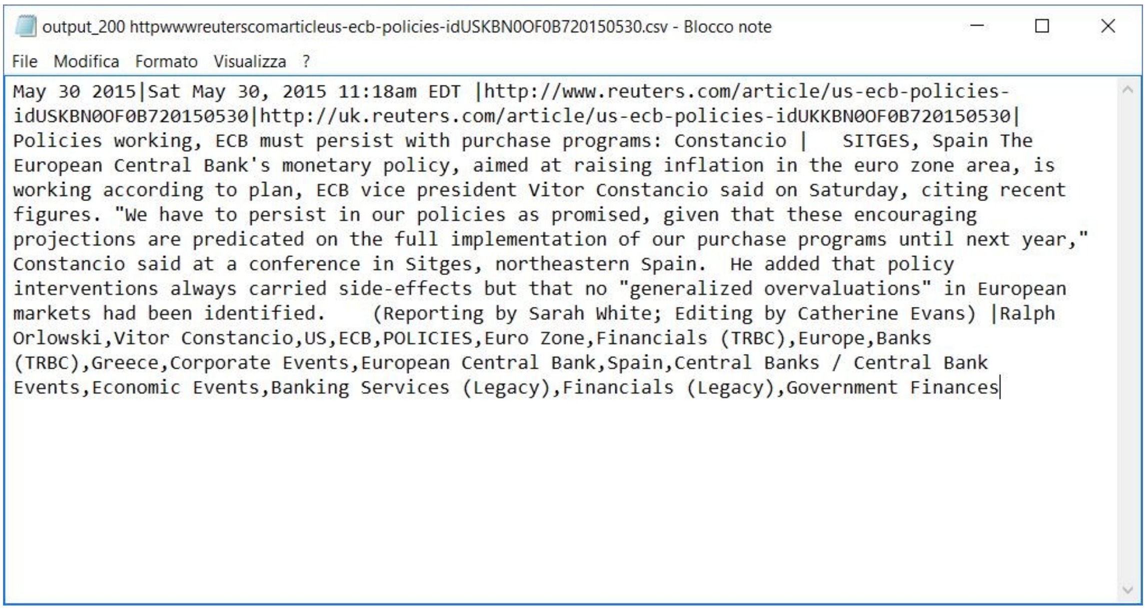
Like · Comment · Share

Source: Authors' calculations based on data from Reuters

5.4 Scraping new articles

At the end of the de-duplication process, there are a little less than 3 million articles for the period 2007-2016. We implement a simple web-scraping algorithm using Scrapy, an open source web crawling framework for Python 2.7.

Figure 32: A scraped article in CSV format (columns are separated by pipes, "—")


output_200 httpwwwreuterscomarticleus-ecb-policies-idUSKBN0OF0B720150530.csv - Blocco note
File
Modifica
Formato
Visualizza
?

```
May 30 2015|Sat May 30, 2015 11:18am EDT |http://www.reuters.com/article/us-ecb-policies-
idUSKBN0OF0B720150530|http://uk.reuters.com/article/us-ecb-policies-idUKKBN0OF0B720150530|
Policies working, ECB must persist with purchase programs: Constancio |   SITGES, Spain The
European Central Bank's monetary policy, aimed at raising inflation in the euro zone area, is
working according to plan, ECB vice president Vitor Constancio said on Saturday, citing recent
figures. "We have to persist in our policies as promised, given that these encouraging
projections are predicated on the full implementation of our purchase programs until next year,"
Constancio said at a conference in Sitges, northeastern Spain.  He added that policy
interventions always carried side-effects but that no "generalized overvaluations" in European
markets had been identified.    (Reporting by Sarah White; Editing by Catherine Evans) |Ralph
Orlowski,Vitor Constancio,US,ECB,POLICIES,Euro Zone,Financials (TRBC),Europe,Banks
(TRBC),Greece,Corporate Events,European Central Bank,Spain,Central Banks / Central Bank
Events,Economic Events,Banking Services (Legacy),Financials (Legacy),Government Finances
```

Source: Authors' calculations based on data from Reuters.

The spider takes as input the list of links obtained in Section 5.2, requests each page containing an article and parses the text to find:

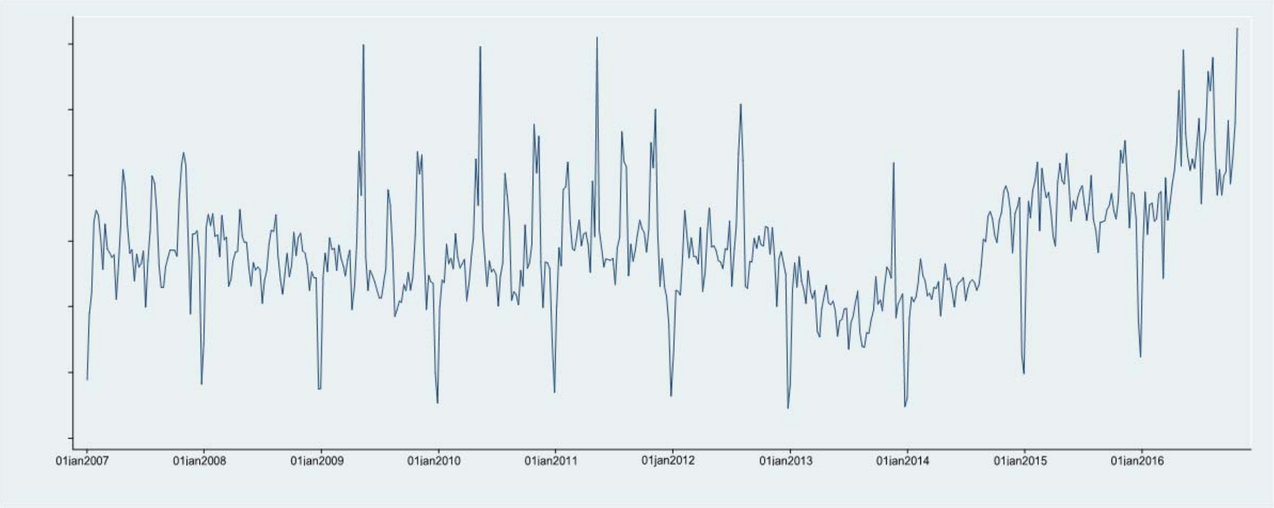
- the date and time of publication,
- the article headline,
- the article body,
- the URL where the article is stored (which usually differs from the URL recorded in the archive),
- an alternative URL (typically, the location of the article in the UK version of the Reuters website),
- the article tags (i.e. a list of keywords about the content of the article, incorporated in the article webpage).

These fields are then written to a single-row CSV file (i.e. one article per file) to make their retrieval easier during the text analysis.

5.5 Article volume over time

Figure 33 provides an overview of the trends in total news article volume. The number of articles published on average each week remains constant until the end of 2014, when it begins to rise. Moreover, the volume of articles drops every last and first week of the year, in correspondence with the Christmas holiday season. This trend is also present in the number of articles that contain the terms used in the estimation of each index. By looking at daily article counts, it is possible to observe that the amount of articles published drops during weekends by about 80% with respect to weekdays.

Figure 33: Weekly volume of article



Source: Authors' calculations based on data from Reuters

5.6 Constructing the uncertainty index

5.6.1 Empirical estimator

Following Eckley (2016), the empirical estimator for uncertainty/risk in our index is the number of articles containing at least one of the terms in the search dictionary over the total amount of articles published in each period:

$$U_t = \frac{m_t}{n_t}$$

where nt is the total number of articles published in the given period, and mt is the number of those articles that express uncertainty or risk.

Eckley (2016) analyses in depth the issue of identification of economic related articles, considering the inclusion of ad hoc to capture economic topics. While relevant in generalistic news outlets, including economy-related keyphrases is not critical if the text data source mostly contains economic related articles. Indeed, Eckley (2016) finds that only 8% of Financial Times articles are not related to the economy and indices computed on a restricted set of articles do not differ significantly from those extracted from the full sample. We have reasons to believe this is also the case for the US edition of the Reuters News website, whose coverage concentrates on business, financial and political topics.

Our attempts to exclude from mt articles about sports by filtering out entries containing tags like "sport", "sports", "Sport" and "SPORT" have highlighted how sport coverage is often associated to economic themes. For example, reports of the exclusion of Russian athletes from the 2016 Olympic Games are associated to economic sanctions against the Russian Federation and news on the football transfer market fall into the wider context of clubs funding availability that depends on the business cycle. For this reason, while we do attempt to exclude sport articles from our sample, we adopt the most conservative filtering criteria: manual comparison of sport news archived under different keywords has determined that the uppercase tag "SPORT" is usually associated with summaries of games results. However, this tag only appears a few dozen times over the entire sample. We do not attempt more stringent filtering techniques as they would dramatically lower article counts as we are using a single source of news data (see also discussion in Eckley (2016)).

5.6.2 From word mentions to indexes

The algorithm works by scanning each article body to detect words specified in a dictionary of terms.[16] For example, the dictionary for the uncertainty index contains the words "uncertainty", "uncertainties" and "uncertain"[17], while the risk index contains "risk", "risks" and "risky". Each time an article is examined by the algorithm its date, URL, tags and file path are written to a csv file, together with an indicator that takes the value 1 if the article contains at least one of the words contained in the dictionary of terms. After all articles have been examined, the code computes daily article frequencies for the total amount of articles and for the articles expressing uncertainty/risk, as well as the actual index U_t, i.e. the ratio between the two.

5.7 Indices

Table 9 summarises the dictionaries of keywords used to identify m_t in each version of the index.

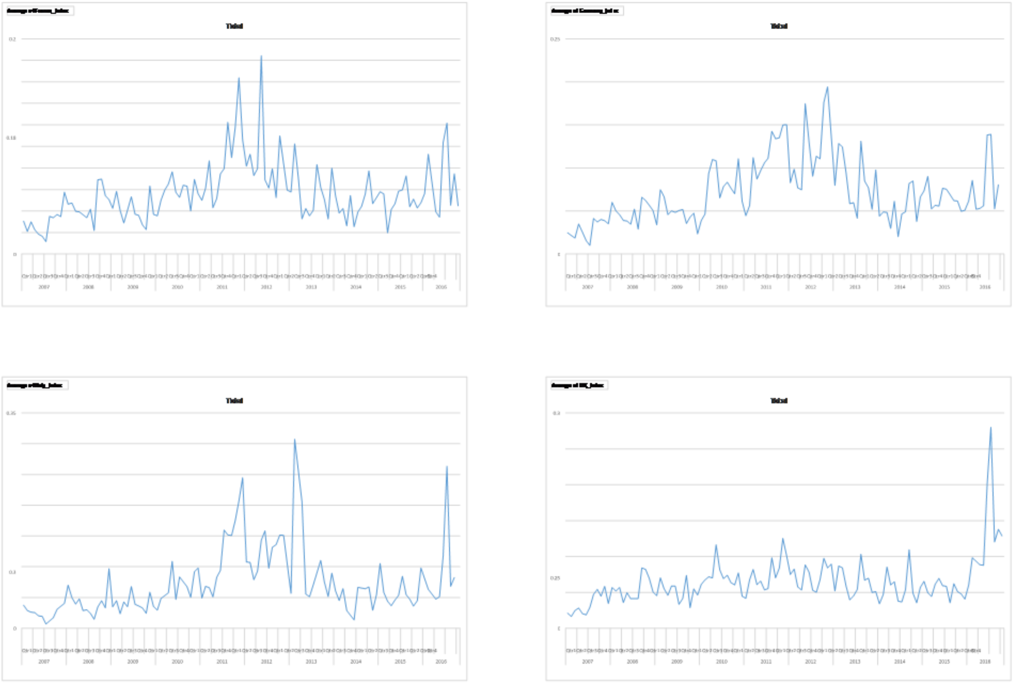
Table 9: Dictionaries of keywords by index

Index	At least one of	AND at least one of
Uncertainty	uncertain, uncertainty, uncertainties	
Risk	risk, risks, risky	
Italy, uncertainty	uncertain, uncertainty, uncertainties	Italy, Italian, Italians
Germany, uncertainty	uncertain, uncertainty, uncertainties	Germany, German, Germans
France, uncertainty	uncertain, uncertainty, uncertainties	France, French
UK, uncertainty	uncertain, uncertainty, uncertainties	UK, Britain, British, United Kingdom, Briton

Source: Authors' calculations based on data from Reuters

Figure 34 reports our main Reuters based uncertainty indexes.

Figure 34: Reuters News Uncertainty Indexes for four countries



Source: Authors' calculations based on data from Reuters

[16] See the provided set of codes.

[17] While Baker et al. (2016) and Alexopoulos and Cohen (2009) do not include "uncertainties" in the dictionary of terms, Eckley (2016) notes that including it increases the number of matched articles without materially changing the semantic range, thus improving the signal-to-noise ratio.

6 Conclusions

In this paper we are concerned with the problem of turning a big, possibly unstructured dataset into a, possibly smaller, structured dataset, which can be later used for economic analysis, in particular for nowcasting and short term forecasting. We provide a general framework for numerical big data, which is also applicable to categorical data once it has been properly transformed.

Several practical examples of the proposed methodology, using either simulated data or publicly available sample data for intraday prices and quotes in financial markets, mobile phone activity, sensor data, online (web-scraped) prices data, online search data, social media data and news articles data. In general, the majority of the examples indicate that linear aggregation seems a suitable choice in numerical exercises, although this largely depends on the conceptual setting of the nowcasting exercise too.

Finally, we specialise the approach to the step-by-step construction of an uncertainty index based on a very large set of news articles, which could be used in future nowcasting and forecasting experiments.

References

1. Abdullah, M.F., Ahmad, K. (2013). “The Mapping Process of Unstructured Data to Structured Data", 3rd International Conference on Research and Innovation in Information Systems -- 2013 (ICRIIS'13).
2. Ailon, N., Chazelle, B. (2006). “Approximate Nearest Neighborhood and the Fast Johnson-Lindenstrauss Transform". Proceedings of the 38st Annual Symposium on the Theory of Computing(STOC), 557-563.
3. Aprigliano, V., Ardizzi, G., Monteforte, L. (2016). “Using the payment system data to forecast the Italian GDP", Bank of Italy, Working Paper.
4. Baker, S.R., Bloom, N., Davis, S.J. (2016). “ Measuring Economic Policy Uncertainty". The Quarterly Journal of Economics, 131(4), 1593-1636.
5. Barndorf-Nielsen, O.E., Hansen, P.R., Lunde, A., Shephard, N. (2009). „Realized kernels in practice: trades and quotes". The Econometrics Journal, 12, C1-C32.
6. Bazzani, A., Giovannini, L., Gallotti, R., Rambaldi, S. (2011). “Now casting of traffic state by GPS data. The metropolitan area of Rome". Conference paper, January 2011. Available at https://goo.gl/kcpPhm.
7. Bholat, D., Stephen H., Pedro S., Schonhardt-Bailey, C. (2015). “Text Mining for Central Banks". Handbooks. Centre for Central Banking Studies, Bank of England.
8. Boutsidis, C., Mahoney, M.W., Drineas, P. (2009). “An Improved Approximation Algorithm for the Column Sum Selection Problem". Proceedings of the 20th Annual SODA, 968-977.
9. Carlsen, M., Storgaard, P.E. (2010). “Dankort payments as a timely indicator of retail sales in Denmark". Danmarks Nationalbank, Working Paper 2010-66.
10. Cavallo, A. (2013). “Online and official price indexes: Measuring Argentina's inflation". Journal of Monetary Economics, 60, 152-165.
11. D. C. M. Initiative (2007). “The Dublin Core Metadata Element Set," Dublin, Ohio: Dublin Core Metadata Initiative.
12. D'Amico, S., Orphanides, A. (2008). “Uncertainty and Disagreement in Economic Forecasting”. Federal Reserve Board Finance and Economics Discussion Series 2008--56.
13. De Meersman, F., Seynaeve, G., Debusschere, M., Lusyne, P., Dewitte, P., Baeyens, Y., Wirthmann, A., Demunter, C., Reis, F., Reuter H.I. (2016). “Assessing the Quality of Mobile Phone Data as a Source of Statistics”. European Conference on Quality in Official Statistics, Madrid, 31 May-3 June 2016.
14. Drineas, P., Mahoney, M.W., Muthukrishnan, S. (2008). “Relative Error CUR Matrix Decompositions", Siam Journal of Matrix Analysis and Applicatons, (30), 844--811.
15. Eckley, P. (2015). “Measuring Economic Uncertainty Using News-Media Textual Data." MPRA Paper, University Library of Munich, Germany.
16. Einav, L., Levin, J. (2013). “The Data Revolution and Economic Analysis". NBER Working Paper No. 19035.
17. Esteves, P.S. (2009). “Are ATM/POS data relevant when nowcasting private consumption?". Banco de Portugal, Working Paper 25-2009.
18. Fernandez-Ares, A.J., Mora, A.M., Arenas, M.G., de las Cuevas, P., Garcia-Sanchez, P., Romero, G., Rivas, V., Castillo, P.A., Merelo, J.J. (2016). “Nowcasting Traffic". Working Paper.
19. Frey, B.J., Dueck, D. (2007). “Clustering by passing messages between data points". Science, 315, 972-976.
20. Galbraith, J.W., Tkacz, G. (2007). “Electronic Transactions as High-Frequency Indicators of Economic Activity". Bank of Canada, Working Paper 2007-58.
21. Galbraith, J.W., Tkacz, G. (2011). “Analyzing Economic Effects of Extreme Events using Debit and Payments System Data". CIRANO Scientific Series, Working Paper 2011s-70.

22. Gandomi, A., Haider, M. (2015). “Beyond the hype: Big data concepts, methods, and analytics". International Journal of Information Management, 35, 134-144.
23. Gentzkow, M., Kelly, B.T., Taddy, M. (2017). “Text as Data". NBER Working Paper No 23276.
24. Henderson, J.V., Storeygard, A., Weil, D.N. (2011). ”A Bright Idea for Measuring Economic Growth". American Economic Review: Papers & Proceedings, 101(3), 194-199.
25. Henderson, J.V., Storeygard, A., Weil, D.N. (2012). “Measuring Economic Growth from Outer Space". American Economic Review, 102(2), 994--1028.
26. Johnson, W., Lindenstauss, J. (1994). “Extensions of Lipschitz Maps into a Hilbert Space". Contemporary Mathematics.
27. Kapetanios, G., Marcellino, M., Papailias, F. (2016). “Big Data and Macroeconomic Nowcasting", Eurostat Working Paper, ESTAT No 11111.2013.001-2015.278.
28. Liu, L.Y., Patton, A.J., Sheppard, K. (2015). “Does anything beat 5-minute RV? A comparison of realized measures across multiple asset classes". Journal of Econometrics, 187, 293-311.
29. Lowe, M. (2014). “Night Lights and ArcGIS: A Brief Guide". MIT Economics Working Paper.
30. Mahoney, M.W., Drineas, P. (2009). ``CUR matrix decompositions for improved data analysis". PNAS, 106(3), 697-702.
31. Mellander, C., Lobo, J., Stolarick, K., Matheson, Z. (2015). “Night-Time Light Data: A Good Proxy Measure for Economic Activity?" PLoS ONE, 10(10).
32. Ng, S. (2016). “Opportunities and Challenges: Lessons from Analyzing Terabytes of Scanner Data". Working Paper.
33. Ordonez, C., Mohaman, N., Garcia-Alvarado, C. (2014). “PCA for large data sets with parallel data summarization". Distributed and Parallel Databases, 32, 377-403.
34. Perduca, V. (2015). “Improving Prediction of Unemployment Statistics with Google trends: Preliminary Experiments", Eurostat Working Paper.
35. Pinkovskiy, M.L. (2013). “Economic Discontinuities at Borders: Evidence from Satellite Data on Lights at Night". MIT Economics Working Paper.
36. Reis, F., Ferreira, P., Perduca, V. (2015). “The Use of Web Activity Evidence to Increase the Timeliness of Official Statistics Indicators", Eurostat Working Paper.
37. Rogers, S. (2016). “What is Google Trends data – and what does it~mean?". Google New Lab. Article available online at: https://goo.gl/ZUHgzY.
38. Rönnqvist, S., Sarlin, P. (2015). “Bank Networks from Text: Interrelations, Centrality and Determinants". Quantitative Finance, 15(10), 1619-1635.
39. Sarlos, T. (2006). “Improved Approximation Algorithms for Large Matrices via Random Projections". Proceedings of the 47 IEEE Symposium on Foundations of Computer Science.
40. Suryadevara, N.K., Mukhopadhyay, S.C., Wang, R., Rayudu, R.K. (2013). “Forecasting the behavior of an elderly using wireless sensors data in a smart home". Engineering Applications of Artificial Intelligence, 26, 2641-2652.
41. Toole, J.L., Lin, Y.-R., Muehlegger, E., Shoag, D., Gonzalez, M.C., Lazer, D. (2015). “Tracking employment shocks using mobile phone data". Journal of the Royal Society Interface, 2015 12 20150185.
42. Venkatasubramanian, S., Wang, Q. (2011). “The Johnson-Lindenstrauss Transform: An Empirical Study". Proceedings of the Thirteenth Workshop on Algorithm Engineering and Experiments, 148--173.
43. Wu, X., Zhu, X., Wu, G.-Q., Ding, W. (2014). “Data mining with big data". IEEE Transactions on Knowledge and Data Engineering, 26(1), 97-107.
44. Zheng, Y., Li, Q., Chen, Y., Xie, X., Ma, W.-Y. (2008). “Understanding Mobility Based on GPS Data". In Proceedings of ACM conference on Ubiquitous Computing (UbiComp 2008), Seoul, Korea. ACM Press: 312-321.

45. Zheng, Y., Xie, X., Ma, W.-Y. (2010). “GeoLife: A Collaborative Social Networking Service among User, location and trajectory". Invited paper, in IEEE Data Engineering Bulletin, 33(2), 32-40.
46. Zheng, Y., Zhang, L., Xie, X., Ma, W.-Y. (2009). “Mining interesting locations and travel sequences from GPS trajectories". In Proceedings of International Conference on World Wild Web (WWW 2009), Madrid Spain. ACM Press: 791-800.

Appendix

In the accompanying program codes we use the following packages:

- “highfrequency" for intraday data,
- “ggmap" to plot the position of user 100 in the Sensor data analysis on Google maps,
- “gtrendsR" to download Google Trends series in R,
- “twitteR" to download Twitter series in R,
- “lubridate" to manipulate dates,
- “apcluster" for APC,
- functions from “clusterv" package,
- “rCUR" for CUR.

Getting in touch with the EU

In person

All over the European Union there are hundreds of Europe Direct Information Centres. You can find the address of the centre nearest you at: http://europa.eu/contact

On the phone or by e-mail

Europe Direct is a service that answers your questions about the European Union. You can contact this service
– by freephone: 00 800 6 7 8 9 10 11 (certain operators may charge for these calls),
– at the following standard number: +32 22999696 or
– by electronic mail via: http://europa.eu/contact

Finding information about the EU

Online

Information about the European Union in all the official languages of the EU is available on the Europa website at: http://europa.eu

EU Publications

You can download or order free and priced EU publications from EU Bookshop at: http://bookshop.europa.eu. Multiple copies of free publications may be obtained by contacting Europe Direct or your local information centre (see http://europa.eu/contact)

EU law and related documents

For access to legal information from the EU, including all EU law since 1951 in all the official language versions, go to EUR-Lex at: http://eur-lex.europa.eu

Open data from the EU

The EU Open Data Portal (http://data.europa.eu/euodp/en/data) provides access to datasets from the EU. Data can be downloaded and reused for free, both for commercial and non-commercial purposes.

KS-TC-17-003-EN-N

Big data conversion techniques including their main features and characteristics

Big data have high potential for nowcasting and forecasting economic variables. However, they are often unstructured so that there is a need to transform them into a limited number of time series which efficiently summarise the relevant information for nowcasting or short term forecasting the economic indicators of interest. Data structuring and conversion is a difficult task, as the researcher is called to translate the unstructured data and summarise them into a format which is both meaningful and informative for the nowcasting exercise. In this paper we consider techniques to convert unstructured big data to structured time series suitable for nowcasting purposes. We also include several empirical examples which illustrate the potential of big data in economics. Finally, we provide a practical application based on textual data analysis, where we exploit a huge set of about 3 million news articles for the construction of an economic uncertainty indicator.

For more information
http://ec.europa.eu/eurostat/

Publications Office

ISBN 978-92-79-70523-6