Time Series Outlier Detection using Metadata and Data Machine Learning in Statistical Production*

B Bogdanova[1], B Ligani[1], A Maurin[1], I Mustafi[1], O Sirello[2]

[1]Bank for International Settlements
[2]Bank for International Settlements, olivier.sirello@bis.org

Abstract

The detection of anomalies in macroeconomic and financial time series is often regarded as a challenging but crucial task to produce high-quality official statistics. This paper proposes a novel, two-step approach that leverages both data and metadata and allows for contextual outlier detection. We apply a mix of unsupervised machine-learning (ML) algorithms which overcome common issues in macroeconomics and financial statistics (e.g. variable length) and ensure suitability for statistical production. While further performance analysis and robustness checks may be needed, our results show that the new method outperforms [substantially] traditional threshold-based rules and highlight its potential to increase data quality and process efficiency.

1 Introduction

The detection of anomalies in macroeconomic and financial time series is often regarded as a challenging but crucial task to produce sound official statistics. Well-designed and performant outlier algorithm checks not only prevent erroneous data to be disseminated but have important implications in terms of productivity for the statistician. Several outlier detection methods exist, ranging from the most traditional ones, based on mean and standard deviation, to more innovative ones leveraging machine-learning (ML) techniques. The latter are particularly useful to cluster time series and evaluate the presence of outliers taking into context the dynamics of the whole cluster.

However, very often, the most common ML algorithms face several challenges when it comes to the analysis of macroeconomic and financial time series. On the one hand, while unsupervised ML techniques are generally preferred for large unlabelled data sets, although some of them may require extensive tuning effort to achieve optimal clustering performance. Conversely, supervised ML algorithms may be easier to calibrate, but it is often very hard to label large and heterogeneous data sets. On the other hand, time series often present variable lengths which might prevent the use of standard distance metrics, such Euclidean, which come by default with the most common ML algorithms.

To face those challenges, we propose a novel approach to detect outliers in macroeconomic and financial time series based on ML techniques. Our approach consists of two steps. First, we cluster time series through metadata and data to identify the *context* against which we perform outlier detection. To do so, we take advantage of time series attributes, such as the title, code and unit, along with the shape of the data. We rely on affinity propagation precomputed with Jaro-Winkler distance matrices as well as density-based spatial clustering of applications with noise (DBSCAN) associated with dynamic time warping (DTW). Second, we perform contextual outlier detection using DBSCAN to identify outliers as those points lying in low-density areas.

*Preliminary version. The views expressed are those of the authors and do not necessarily reflect those of the Bank for International Settlements. All errors are our own.

2 Background and motivation

The BIS Data Bank is a data warehouse hosting more than sixty thousand macroeconomic and financial time series, most of which are submitted by central banks to the Bank for International Settlements (BIS). Outliers are in principle rare as data quality checks are performed by reporting central banks. However, it may occur that some observations are flagged as anomalies and prompt manual inspection by BIS statisticians.

Outlier checks currently in place in the BIS Data Bank identify outliers when the absolute difference between the observation i and the moving average MM over the rolling window of length k is greater or equal to its standard deviation σ over the window of same length k multiplied by a factor b:

$$\left|i - MM\left(i\right)_k\right| \geq \sigma\left(i\right)_k \times b \tag{1}$$

This method presents several shortcomings. First, it defines point outliers against predefined thresholds which are often not suited for time series with linear breaks, such as financial time series. Furthermore, it does not allow for contextual outlier detection. For example, when a financial shock propagates across countries, related domestic time series are expected to feature the same jumps, albeit possibly with different orders of magnitude. Performing outlier detection on each individual series may lead to flag these jumps as outlying values. Conversely, taking into account also the related time series, these jumps are expected to be treated as inliers since it is unlikely that all related series, reported by multiple central banks, share the same outliers over the same time window. The BIS Data Bank hosts this information as it typically gathers cross-country data on related indicators from the reporting central banks.

Contextual anomalies are those "points which can be normal in a certain context, while detected as anomaly in another context" (Braei and Wagner 2020). They differ from point anomalies which correspond to data points deviating significantly from the rest of the data sample. For instance, the observation x for period t can be flagged as point outlier if it deviates significantly from other neighbouring observations $[x_{t-n}, x_{t+n}], n \in \mathbb{N}$. Under contextual outlier detection, the same observation x_t is checked against $[x_{t-n}, y_{t_n}, z_{t-n}, ..., x_{t+n}, y_{t+n}, z_{t+n}], n \in \mathbb{N}$ where y and z are the neighbouring observations of the *related* time series. Contextual outlier detection is very often considered challenging for at least two reasons. First, it requires the identification of the context, for example the *related* time series. Secondly, when not properly tuned, contextual outlier detection methods may fail to identify anomalies that would have been identified by individual outlier detection.

Against this background, we propose a new method relying on ML that performs outlier detection taking into account also related time series. Our method has two main steps. First, the time series are clustered based on their metadata and data. Second, contextual outlier detection is performed for each cluster. Our proposal aims to improve the current statistical production pipeline for the BIS Data Bank.

3 Clustering metadata and data

3.1 Metadata

This section briefly describes the approach we propose to cluster the times series based on their metadata.

3.1.1 Clustering the BIS Data Bank Topics

The BIS Data Bank contains time series which feature metadata stored as attributes. Time series are grouped by statistical domains or *blocks* which are further broken down into categories or *topics.* A 4-four letter code, or *topic code*, and a mandatory and standardized text description, or *topic title*, describe the topic. The topic code is a useful starting point for clustering time series based on metadata since we reasonably assume that series falling under the same topic should belong to the same metadata cluster. However, the level of granularity of the topic appears to be too detailed for our purpose. For example, we expect that series on *Interest rate, policy rate* are very close to series related to *Interest rate, official discount rate* albeit they belong to two different topics. For this reason we propose to leverage unsupervised ML techniques to cluster those series based on not only on the topic code but *also* their mandatory description.

A key choice is the distance metric to calculate the similarity between strings. Widely-used distance functions are *edit distances* such the Levensthein distance which measures the smallest number of operations (i.e. insertion, deletion, substitution) needed to make identical a pair of strings. However, it typically fails to take into account the order of the common characters, a feature which is key in our case. We use a variant of the Jaro distance (Jaro 1989) which computes the number of common characters and transpositions, taking also into account the agreeing initial characters between strings (Winkler 1999). This approach accomodates our purpose to compare the concatenation of the topic code, which is located at the beginning of the string, with the topic description. Also, Jaro-Winkler has been found to perform well on short strings which is also our case (Christen 2006).

We calculate the similarity matrix with the Jaro-Winkler distance across all pairs of topics. We then feed the affinity propagation algorithm with the similarity matrix in order to derive the clusters based on metadata. Affinity propagation is a purely data-driven and unsupervised clustering algorithm, eg it does not require to set *ex ante* the number of clusters (Frey and Dueck 2007). It clusters data points by grouping points (i) around those "highly representative" or "exemplars" (k). It does so by iteratively assessing the responsibility (r, step 1), similarity (s) and availability (a), while maximising s and a (steps 2 and 3):

$$r(i,k) = s(i,k) - max_{k' \neq k}\{a(i,k') + jarow(i,k')\}$$

$$a(i,k)_{i \neq k} = min\Big(0, r(k,k) + \sum_{i' \not\subset \{i,k\}} max(0, r(i',k))\Big)$$

$$a(k,k) = \sum_{i' \neq k} max(0, r(i',k))$$

3.2 Data

This section briefly describes the algorithms we propose in this paper to cluster the time series based on their shape. This step allows us to further refine the *pre*-clustering of time series based on their metadata.

3.2.1 Clustering unequal-length time series through Dynamic Time Warping

The BIS Data Bank contains an extensive volume of variable-length time series. Often, different series may reflect the same phenomenon shifted over time. For example, a financial shock may propagate from one country to another with some lag. Assuming that two time series a and b measure the same financial indicator, time-fixed distances (i.e. distances which compare only values measured at the same point in time) may show a greater distance between a and b than distances which compare the actual shape of the two time series. As a result, for many cases, it may be desirable to group time series based on their shape. Besides, series featuring the same frequency or spaced at uniform time intervals may also have missing data, for example due to data gaps, and may feature other distortions.

The distance function used to cluster time series is a crucial choice for any clustering method as it has serious impact on its overall accuracy and performance. Standard distance functions usually associated with time series partitioning assume a fixed mapping between sequences and follow the temporal order of the data points (Wang et al. 2013; Guijo-Rubio et al. 2020). The main reason is that time series, unlike other data representations, are sequential-based arrays for which the order of observations carries relevant information. Among those metrics, the Euclidean distance function is one of the most popular, along with Manhattan and Minkowski. However, those methods cannot handle series with unequal length and do not account for time shifts in the data (Ding et al. 2008).

In order to calculate the distance between variable-length time series, we use dynamic time warping - DTW (Berndt and Clifford 1994). This algorithm is one of the most widely used for time series classification as it allows to calculate the distance between arrays with different lengths, thus getting their similarity based on their shapes and reveal their temporal dynamics (Bagnall et al. 2016). Without loss of generality, we can assume that X and Y are time series of length N and M, with $M \geq N$:

$$X = [x_1, x_2, ..., x_i, x_N]; Y = [y_1, y_2, ..., y_j, y_M], , N and M \in \mathbb{N}$$

the two time series can be represented in a matrix $C = [c_{ij}]_N^M$, where $c_{ij} = d(x_i, y_j)$ and d is the Euclidean distance. The DTW algorithm will find the *warping path*, which is defined by the correspondences $(i_k, j_k)_{k=1}^M$ of elements x_{i_k} in X to elements y_{j_k} in Y so that:

- x_1 corresponds to y_1 and x_N to y_M
- the time ordering of X and Y is preserved, i.e. $i_1 \leq i_2 \leq ... \leq i_N$ and $j_1 \leq j_2 \leq ... \leq j_M$
- there are no jumps, i.e. $i_{k+1} - i_k$ and $j_{k+1} - j_k$ are 0 or 1
- the cumulative distance of each mapped pairs $\Sigma_{k=1}^M d(x_{i_k}, y_{j_k})$ is the minimum possible.

The DTW distance is the cumulative distance of the mapped pairs, such as

$$DTW(X, Y) = \Sigma_{k=1}^M d(x_{i_k}, y_{j_k}) \tag{2}$$

Since DTW forces time series to be stretched or compressed, it may also create wrong matching and distort the true similarity between time series (Keogh and Ratanamahatana 2005; Ding et al. 2008). One possible solution to prevent pathological mappings is to apply a cap on the maximal shift that the warping path is allowed from the diagonals, for instance using a Sakoe-Chiba band (Dau et al. 2018; Sakoe and Chiba 1978). Also, in our case, because time series are already clustered based on their metadata, we reasonably expect this risk to be low.

3.2.2 Clustering with Density-based spatial clustering of applications with noise (DBSCAN)

To cluster time series, we rely on Density-Based Spatial Clustering of Applications with Noise (DBSCAN) which we feed with the distance matrix computed via the DTW algorithm as described in the previous section. DBSCAN is an algorithm which discovers clusters on the areas with the highest density, while low-density areas are associated with noise. The formal notion of “clusters” and “noise” is straightforward in Ester et al. (1998). To form a cluster, the neighbourhood of a certain radius shall contain a minimum number of points ($minPts$). Considering the distance function $dist(x, y)$ between points x and y in a database of points D, the ϵ-neighbourhood of x is defined by $\{q \in D | dist(x, y) \leq \epsilon\}$. In our case, x and y are time series, $dist(x, y)$ is given by DTW, while ϵ and $minPts$ are set, respectively, to the median value of each cluster - excluding observations equal to 0 - and 2. This means that the ϵ-neighbourhood of a given point p has to contain at least 2 points in order to form a cluster.

For our purpose, DBSCAN brings several advantages. First, it is both data-driven and unsupervised. For statistical production we consider this feature to be highly critical in order to generalize our method in a fully automated pipeline. In this respect, it shall not require any manual intervention or domain-specific knowledge to set the input parameters which partitioning and hierarchical algorithms often necessitate (Kaufman and Rousseeuw 2009). Furthermore, DBSCAN associated with DTW has also be proven to perform relatively well to cluster time series with noise, which is often the case for real time series (Lyudmyla Kirichenko 2019).

4 Outlier Detection

This section goes over the steps that we follow in order to perform outlier detection on the clustered series.

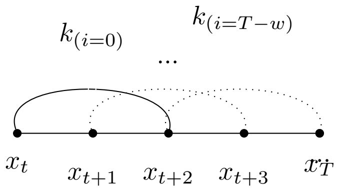
4.0.1 Contextual outlier detection

Our key goal is to perform point outlier detection on time series taking into account points from related time series. Since time series pooled in the same cluster share the same shape and are expected to carry the information of related statistical phenomena, we perform outlier detection over the data points of all series belonging to the same cluster. The intuition is straightforward: assuming that the 10-year government French bond yield faces a sharp rise which is not explained by an idiosyncratic shock, we may expect to find the same rise across other euro area countries, albeit with differences in level. Hence, we may find a point to be an outlier while checking the single series, whereas we would flag the same point as inlier by looking at the whole cluster of time series featuring the same jump.

Definition 4.1. Data point x_{t,S_1} is flagged as outlier if it deviates significantly from its neighbouring points, $x_{t-1,S_1}, ..., x_{t-n,S_1}$ as well as from the data points belonging to related series $x_{t-1,S_2}, ..., x_{t-n,S_Z}$.

4.0.2 DBSCAN

We rely on DBSCAN to capture outliers, ie those data points that lie in low-density areas and, consistently, we expect inliers to lie in high-density ones. We formalise this intuition as follows: we detect outliers for each region of data D which contains the data points lying in cluster C and within the sliding window k of size w up to when the number of iterations (i) is equal to the difference between the total number of periods T and the window size



Because differences in levels may exist and are not relevant in our case, we differentiate the time series and apply a *min-max* scaler. The latter typically scales data points between 0 and 1, i.e. if $x = (x_t)_{t=1}^{N}$, then the *min-max* scaled series x' is given by:

$$x'_t = \frac{(x_t - \min x)}{(\max x - \min x)} \times (a - b) + b \tag{3}$$

where tipically $a = 1$ and $b = 0$. We apply those transformations for each time series.

We follow a conservative approach to define ϵ and consider that at least 2 points are needed to form a cluster (ie. $minPts = 2$). Outliers are those data points which are part of the clusters which contain the minimum number of data points, including those which could not be clustered (i.e. noise). We specify m as the m clusters featuring the minimum data points clustered, including noise. For example, let's assume that DBSCAN returns four clusters for iteration i, C_1, C_2, C_3 and C_4 respectively with sizes s_1, s_2, s_3, s_4. If m equals 2 and s_1 and s_2 are lower than s_3 and s_4, then data points belonging to C_1 and C_2 are flagged as outliers. These parameters help to detect the presence of multiple outliers in D whose cluster sizes are different, but smaller than clusters containing inliers.

5 Application

In this section, we present a brief empirical test to illustrate the methods outlined in the previous sections.

5.1 Data sample

For the purpose of this paper, we only consider a small subset of the BIS DataBank time series. Series are organized in topic codes which describe the underlying statistical data. A topic code is typically made of 4 letters and is associated to a unique statistical phenomenon. The order of the letters in the topic code has a precise meaning: the initial letter refers to the broadest category while the other ones relate to further breakdowns. For example, topic code *HBBA* contains series related to *Interest rate, official, discount rate/base rate*, which belongs to block *H* (i.e. first letter) on *interest rate*. The characteristics of the sample are summarized below:

Table 1: Data sample

Time series (n)	668
Topic codes (n)	43
Frequency	M
Start date	January 1919
End date	May 2023
Missing observations	1.99%
Length	359

Missing observations correspond to the average share of *NaN* values, that is the data gaps, per series. Consistently, the length refers to the average number of periods, i.e. months, per each time series. However, start and end dates are the minimum and maximum dates across the whole data sample.

5.2 Time series metadata and data clustering

In order to define the *context* against which we perform outlier detection, we cluster time series leveraging both metadata and data. We start by clustering the series by taking three metadata, namely the *topic code*, the *topic name* and the *unit code*. Those attributes play a determinant role for our analysis: the topic code is expected to give an indication about the classification of the series. However, the code alone is not enough to cluster the time series, since there may be other topic codes closely related. For example, the series in *HBBA* (*Interest rate, official, discount rate/base rate*) are closely related to codes *HBAA* (*interest rate official discount rate base rate*), *HBDA* (*interest rate official deposit facility*), *HBEA* (*interest rate official rate advances*) and so on. Hence, we look at the topic name on top of the code. Finally, the unit code is also important, as to make sure that series falling in the same cluster share the same unit of measure.

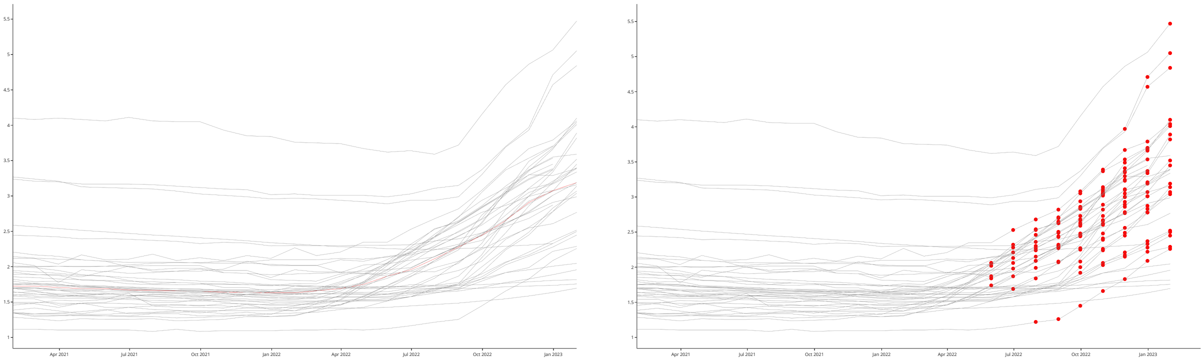
We use affinity propagation with the Jaro-Winkler distance function (Section 3.1). We preprocess the metadata in order to reduce errors or biased results, for example by removing stopwords and punctuation, before calculating the distances for each combination of topic codes and names. We feed Affinity Propagation with the distance matrix and we chose a low damping value to prevent overshooting. We set the preference to each point to be 150% of the median similarity. This helps us to reduce the number of clusters and obtain a better distribution of cluster sizes. We create groups from the clusters assigning to each of them a name by appending the unit code of clustered time series to a sequential positive integer. In this respect, two series falling under the same cluster but with two different units are assigned to two different groups.

As a second step, we further refine these results by clustering the time series based on their shape. To do so, we use dynamic time warping and DBSCAN (Section 3.2). More specifically, we first differentiate each time series before calculating the warping paths. To lower the risk of pathological mappings between time series, we also apply a window equal to 3, i.e. one month can be mapped up to three months, which sounds to be reasonable for our data. We use the distance matrix calculated through DTW to cluster the series using DBSCAN. With low tuning effort, we manage to cluster 62.28% of the series. This result might be further improved by applying other preprocessing methods, including smoothing, although preliminary tests do not show significant gains. Overall, the results sound promising: for example, the cluster *7C3681* contains 46 series relating to the topics: *interest rate total loans total amounts outstanding, interest rate total loans new loans, interest rate mortgage construction loans, interest rate new mortgage loans.*

5.3 Contextual outlier detection

Once we retrieve the clusters for the time series in our sample, we check the presence of outliers using DBSCAN. As explained in Section 4, we differentiate and apply a *min-max* scaler between 0 and 1 for each time series before clustering. We run the model by setting ϵ to be equal to the 0.02 quantile of the sample, and iterate the checks over a sliding window of size 12, i.e. a year. Sample data is expected to be without outliers.

The comparison between the contextual and the current check is striking. Whenever there are exogenous shocks that affect the whole context, contextual outlier detection does not raise any false positive. Conversely, because the current outlier check does not take into account the context, it flags outliers over the same period. We show an example of this finding below related to the time series on loans and mortgages interest rates during the recent sharp rise in interest rates since mid-2022.



(a) Contenxtual

(b) Current

Figure 1: Comparison between contextual and current outlier checks during the sharp increase in interest rates since mid-2022

6 Conclusion and way forward

We showed that clustering time series based on their metadata and data is a powerful tool to enhance traditional outlier detection checks, such as the current one in the BIS DataBank. Relying on unsupervised ML techniques, we are able to derive the context against which performing outlier detection and prevent false positives at the level of each individual series when an exogenous shock affect the whole sample.

Our method is still in its early stages and requires more research in three directions. First, we need to better assess the mix of individual and contextual outlier checks. In fact, contextual outlier detection may be less sensitive to individual point outliers that may occur across the individual time series, hence requires appropriate tuning before being implemented in production. Also, further testing and robustness checks may help to detect the presence of false negatives which may lower the data quality of the series produced. Secondly, more tuning is also required to test different input parameters of the ML algorithms referenced in this paper. This is particularly the case for DBSCAN since small changes in its parameters may yield significantly different results. Finally, as we aim at introducing this check in the context of statistical production, a careful analysis of performance needs also to be weighted-in along with the design of a ML-based pipeline.

References

Bagnall, Anthony J., Aaron Bostrom, James Large, and Jason Lines. 2016. "The Great Time Series Classification Bake Off: An Experimental Evaluation of Recently Proposed Algorithms. Extended Version." *CoRR* abs/1602.01711. http://arxiv.org/abs/1602.01711.

Berndt, Donald J., and James Clifford. 1994. "Using Dynamic Time Warping to Find Patterns in Time Series." In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, 359–70. AAAIWS'94. Seattle, WA: AAAI Press.

Braei, Mohammad, and Sebastian Wagner. 2020. "Anomaly Detection in Univariate Time-Series: A Survey on the State-of-the-Art." *arXiv Preprint arXiv:2004.00433*.

Christen, Peter. 2006. "A Comparison of Personal Name Matching: Techniques and Practical Issues." In *Sixth IEEE International Conference on Data Mining-Workshops (ICDMW'06)*, 290–94. IEEE.

Dau, Hoang Anh, Diego Furtado Silva, François Petitjean, Germain Forestier, Anthony Bagnall, Abdullah Mueen, and Eamonn Keogh. 2018. "Optimizing Dynamic Time Warping's Window Width for Time Series Data Mining Applications." *Data Mining and Knowledge Discovery* 32: 1074–1120.

Ding, Hui, Goce Trajcevski, Peter Scheuermann, Xiaoyue Wang, and Eamonn Keogh. 2008. "Querying and Mining of Time Series Data: Experimental Comparison of Representations and Distance Measures." *Proc. VLDB Endow.* 1 (2): 1542–52. https://doi.org/10.14778/1454159.1454226.

Frey, Brendan J, and Delbert Dueck. 2007. "Clustering by Passing Messages Between Data Points." *Science* 315 (5814): 972–76.

Guijo-Rubio, David, Antonio Manuel Durán-Rosal, Pedro Antonio Gutiérrez, Alicia Troncoso, and César Hervás-Martínez. 2020. "Time-Series Clustering Based on the Characterization of Segment Typologies." *IEEE Transactions on Cybernetics* 51 (11): 5409–22.

Jaro, Matthew A. 1989. "Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida." *Journal of the American Statistical Association* 84 (406): 414–20.

Kaufman, Leonard, and Peter J Rousseeuw. 2009. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons.

Keogh, Eamonn, and Chotirat Ann Ratanamahatana. 2005. "Exact Indexing of Dynamic Time Warping." *Knowledge and Information Systems* 7: 358–86.

Lyudmyla Kirichenko, Anastasiia Tkachenko, Tamara Radivilova. 2019. "Comparative Analysis of Noisy Time Series Clustering." In *International Conference on Computational Linguistics and Intelligent Systems*.

Sakoe, H., and S. Chiba. 1978. "Dynamic Programming Algorithm Optimization for Spoken Word Recognition." *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26 (1): 43–49. https://doi.org/10.1109/TASSP.1978.1163055.

Sander, Jörg, Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu. 1998. "Density-Based Clustering in Spatial Databases: The Algorithm Gdbscan and Its Applications." *Data Mining and Knowledge Discovery* 2: 169–94.

Wang, Xiaoyue, Abdullah Mueen, Hui Ding, Goce Trajcevski, Peter Scheuermann, and Eamonn Keogh. 2013. "Experimental Comparison of Representation Methods and Distance Measures for Time Series Data." *Data Mining and Knowledge Discovery* 26: 275–309.

Winkler, William E. 1999. "The State of Record Linkage and Current Research Problems." *Statistical Research Division, US Bureau of the Census, Wachington, DC.*