Methods for Automatic Record Matching and Linkage and their Use in National Statistics

Leicester Gill
Oxford University
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreword</td>
<td>7</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>9</td>
</tr>
<tr>
<td>Members of the GSS(M) Task Force on Data Sharing and Matching</td>
<td>10</td>
</tr>
<tr>
<td>Executive Summary</td>
<td>11</td>
</tr>
<tr>
<td>1. Introduction and aims</td>
<td>13</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>13</td>
</tr>
<tr>
<td>1.2 Aims of the report</td>
<td>14</td>
</tr>
<tr>
<td>1.3 Outline of the report</td>
<td>14</td>
</tr>
<tr>
<td>2. Development of record matching and linkage</td>
<td>16</td>
</tr>
<tr>
<td>2.1 What is record matching and linkage?</td>
<td>16</td>
</tr>
<tr>
<td>2.2 History of record matching and linkage</td>
<td>16</td>
</tr>
<tr>
<td>2.2.1 Early developments in record matching</td>
<td>16</td>
</tr>
<tr>
<td>2.2.2 Advent of computing systems for information processing</td>
<td>17</td>
</tr>
<tr>
<td>2.2.3 Development of probabilistic methods of record matching</td>
<td>18</td>
</tr>
<tr>
<td>2.2.4 Introduction of phonetic codes into record matching</td>
<td>19</td>
</tr>
<tr>
<td>2.2.5 Development of general purpose matching and linking software</td>
<td>20</td>
</tr>
<tr>
<td>2.3 Record linkage applications in the UK</td>
<td>20</td>
</tr>
<tr>
<td>2.4 Uses of record matching and linkage</td>
<td>23</td>
</tr>
<tr>
<td>3. An introduction to record matching methods</td>
<td>27</td>
</tr>
<tr>
<td>3.1 Exact matching (deterministic, all-or-none methods)</td>
<td>27</td>
</tr>
<tr>
<td>3.2Probabilistic matching</td>
<td>28</td>
</tr>
<tr>
<td>3.3 Other record matching algorithms used for record linkage and text retrieval</td>
<td>30</td>
</tr>
<tr>
<td>4. Practical procedure for record matching and linkage</td>
<td>31</td>
</tr>
<tr>
<td>4.1 The pre-matching processes</td>
<td>31</td>
</tr>
<tr>
<td>4.2 Select the matching method</td>
<td>35</td>
</tr>
<tr>
<td>4.3 Match the file</td>
<td>35</td>
</tr>
<tr>
<td>4.4 Link the matched records to the rest of the file</td>
<td>36</td>
</tr>
<tr>
<td>4.5 Build the linked file and check for errors</td>
<td>36</td>
</tr>
<tr>
<td>4.6 Other considerations</td>
<td>37</td>
</tr>
<tr>
<td>5. Data editing and parsing for matching and linkage</td>
<td>38</td>
</tr>
<tr>
<td>5.1 Selection and definition of matching variables</td>
<td>38</td>
</tr>
<tr>
<td>5.1.1 Selection of the matching variables</td>
<td>38</td>
</tr>
<tr>
<td>5.1.2 Typical errors that occur in matching variables</td>
<td>40</td>
</tr>
<tr>
<td>5.2 Editing, parsing and standardisation of the matching variables</td>
<td>44</td>
</tr>
<tr>
<td>5.2.1 Parsing and standardisation of the matching variables</td>
<td>44</td>
</tr>
<tr>
<td>5.2.2 Creation of linkage files</td>
<td>48</td>
</tr>
<tr>
<td>5.3 Blocking and sorting of the data files</td>
<td>51</td>
</tr>
<tr>
<td>5.4 The structure and organisation of files to be record matched</td>
<td>53</td>
</tr>
<tr>
<td>6. Exact or deterministic methods of record matching</td>
<td>54</td>
</tr>
<tr>
<td>6.1 Exact methods of record matching</td>
<td>54</td>
</tr>
<tr>
<td>6.1.1 Defining a unique identifier</td>
<td>54</td>
</tr>
<tr>
<td>6.1.2 Blocking, sorting and matching</td>
<td>55</td>
</tr>
<tr>
<td>6.1.3 Checking the validity of the match</td>
<td>55</td>
</tr>
<tr>
<td>6.1.4 Resolving uncertainties</td>
<td>55</td>
</tr>
<tr>
<td>6.2 Risks associated with wrong matching</td>
<td>56</td>
</tr>
<tr>
<td>6.3 Typical match rates</td>
<td>56</td>
</tr>
</tbody>
</table>
7. Probabilistic methods of record matching
   7.1 When is probabilistic record matching used?
   7.2 Basic concepts of probabilistic matching
   7.3 From frequency ratios to match weights
   7.4 Combination of weights over all variables in the identifying set
   7.5 Generation of outcome specific weights
   7.6 Blocking the file to reduce the number of unproductive matches
   7.7 File blocking and matching where the variables used for the blocking
       keys to exhibit some error
   7.8 Constraints on matching to reduce the number of unproductive
       comparisons
   7.9 Determination of the match acceptance threshold
      7.9.1 Matching the record pairs
   7.10 Setting the matching threshold
      7.10.1 Preliminary decisions on the linkages – the formal
           Fellgi-Sunter theory
      7.10.2 Threshold setting and Type I and Type II errors
   7.11 Resolution of uncertainties: how to deal with the queries and
       errors in matching
   7.12 Consequences of errors in matching, Type I and Type II errors
   7.13 Combining results from many match runs using different blocking keys
   7.14 Reducing risks associated with wrong matching
   7.15 Typical match rates

8. Methods of building the linked file from the matched records
   8.1 Building the linked files
   8.2 Sorting and logically checking the records
   8.3 Other outputs from the matching process

9. Commercial software packages available for record linkage

10. Typical results from the selected GSS linkage projects

11. Future developments in record linkage
    11.1 Use of non-invasive physical characteristics of a person
    11.2 Upgrade of the parsing, editing and blocking routines
    11.3 Matching software based on rule-based search methods being
        developed for the Internet
    11.4 Use of intelligent and expert methods

12. Ethics, confidentiality and data protection issues in record linkage

13. Conclusions and Recommendations

Annexes
   Annex A Fellgi-Sunter theory of probabilistic record matching and linkage
   Annex B Phonetic coding systems
   Annex C Other record matching algorithms
   Annex D Use of check characters for internal checking of numbering systems
   Annex E Distribution of names in England and Wales from analysis of
       the NHSCR
   Annex F The structure and organisation of files to be record matched
   Annex G Details of the selected GSS linkage projects

Glossary and abbreviations
   Glossary
   Abbreviations

References and bibliography
<table>
<thead>
<tr>
<th>Box</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Key technical issues in the development of record matching</td>
<td>19</td>
</tr>
<tr>
<td>2.2</td>
<td>Applications of record linkage in health studies</td>
<td>24</td>
</tr>
<tr>
<td>2.3</td>
<td>Some applications of record linkage to other types of National Datasets</td>
<td>25</td>
</tr>
<tr>
<td>4.1</td>
<td>Stages for record matching and linkage</td>
<td>32</td>
</tr>
<tr>
<td>5.1</td>
<td>Examples of variables that are set to unknown values</td>
<td>42</td>
</tr>
<tr>
<td>5.2</td>
<td>Examples of names or titles that have low discriminating power and will therefore generate low outcome weights</td>
<td>43</td>
</tr>
<tr>
<td>5.3</td>
<td>Examples of titles that are usually embedded in or appended to the surname or forename variables</td>
<td>51</td>
</tr>
<tr>
<td>5.4</td>
<td>A sample of a Lexicon for converting forenames to a formal name</td>
<td>45</td>
</tr>
<tr>
<td>5.5</td>
<td>Some typical variations on the spelling of ABIGAIL</td>
<td>46</td>
</tr>
<tr>
<td>5.6</td>
<td>Creating extra master file records where there are many variations of the surnames and the forenames</td>
<td>50</td>
</tr>
<tr>
<td>5.7</td>
<td>Details of the ORLS record headers</td>
<td>51</td>
</tr>
<tr>
<td>6.1</td>
<td>Typical results obtained from exact matching datasets containing names or other matching variables</td>
<td>57</td>
</tr>
<tr>
<td>7.1</td>
<td>Cases where exact and probabilistic matching are appropriate</td>
<td>58</td>
</tr>
<tr>
<td>7.2</td>
<td>Global frequency ratios for agreements and disagreements of selected identifiers on matched pairs of records</td>
<td>61</td>
</tr>
<tr>
<td>7.3</td>
<td>Frequencies of specific values of identifiers</td>
<td>61</td>
</tr>
<tr>
<td>7.4</td>
<td>Binit weights for some common identifiers</td>
<td>61</td>
</tr>
<tr>
<td>7.5</td>
<td>Probability ratios for outcomes based on day of birth and month of birth</td>
<td>67</td>
</tr>
<tr>
<td>7.6</td>
<td>Relative characteristics of identifying variables used for person matching in health-related datasets</td>
<td>65</td>
</tr>
<tr>
<td>7.7</td>
<td>Calculation of outcome specific weight for a surname</td>
<td>68</td>
</tr>
<tr>
<td>7.8</td>
<td>Calculation of outcome specific weight for a forename</td>
<td>69</td>
</tr>
<tr>
<td>7.9</td>
<td>Calculation of outcome specific weight for a date of birth</td>
<td>70</td>
</tr>
<tr>
<td>7.10</td>
<td>The outcome specific weights used by the ORLS for matching</td>
<td>71</td>
</tr>
<tr>
<td>7.11</td>
<td>ORLS string comparison method</td>
<td>74</td>
</tr>
<tr>
<td>7.12</td>
<td>Winkler and Jaro string comparator</td>
<td>74</td>
</tr>
<tr>
<td>7.13</td>
<td>Comparison of two strings using Bigrams</td>
<td>74</td>
</tr>
<tr>
<td>7.14</td>
<td>Typical results obtained from probabilistic matching datasets containing names or other matching variables</td>
<td>88</td>
</tr>
<tr>
<td>8.1</td>
<td>Part outputs from the matching runs in the ORLS system</td>
<td>90</td>
</tr>
<tr>
<td>8.2</td>
<td>Sequence of records with same Person Number</td>
<td>91</td>
</tr>
<tr>
<td>8.3</td>
<td>Sequence of records for a person who has two Person Numbers</td>
<td>92</td>
</tr>
<tr>
<td>8.4</td>
<td>Sequence of records for a person who might have three or more different Person Numbers</td>
<td>93</td>
</tr>
<tr>
<td>8.5</td>
<td>Examples of ONCA coding</td>
<td>93</td>
</tr>
<tr>
<td>8.6</td>
<td>Types of error that can occur in numbering systems</td>
<td>118</td>
</tr>
<tr>
<td>8.7</td>
<td>The error detection rate in self checking numbers using the remainder method of check character derivation</td>
<td>119</td>
</tr>
<tr>
<td>8.8</td>
<td>Computation of the check digit in the new NHS number</td>
<td>120</td>
</tr>
<tr>
<td>8.9</td>
<td>Frequency of surnames on the NHSCR</td>
<td>121</td>
</tr>
<tr>
<td>8.10</td>
<td>The top 20 surnames on the NHSCR (both sexes all ages)</td>
<td>122</td>
</tr>
<tr>
<td>8.11</td>
<td>The top 20 forenames on the NHSCR (all ages)</td>
<td>123</td>
</tr>
</tbody>
</table>
# List of figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 4.1</td>
<td>Stages for record matching and linkage</td>
<td>33</td>
</tr>
<tr>
<td>Figure 7.1</td>
<td>Frequency distribution of Binit weights for pairs of records</td>
<td>79</td>
</tr>
<tr>
<td>Figure 7.2</td>
<td>Sample portion of the threshold acceptance array showing the number of matches and non-matches by outcome weight for names (x axis) and non-names variables (Y axis)</td>
<td>83</td>
</tr>
<tr>
<td>Figure 7.3</td>
<td>A sample array used for matching hospital records with hospital records</td>
<td>84</td>
</tr>
</tbody>
</table>
Foreword

This review, of current best practice methodology for data matching by Leicester Gill, was commissioned by the methodology subgroup of the Government Statistical Service Task Force on record matching and data sharing which was set up in June 1999.

It brings together the concepts and practicalities of applying matching methodologies to data linkage, and should be a valuable source for government statisticians. There is an ever-increasing demand from users of statistics for timely and high quality statistics, when at the same time statistical agencies face growing pressure on their resources. This requires looking at sources of data supplementary to traditional ones - censuses and surveys. Administrative data, routinely collected through normal day-to-day government activities, are a rich source of statistical information hitherto under-exploited. Where privacy concerns allow it, and subject to rigorous confidentiality constraints, linking data facilitates better use of the information that government currently have available.

The numerous benefits of record linkage for official statistics include the compilation of new statistical outputs, improving the quality of existing outputs, and reducing respondent burden and costs.

The purpose of this report is to serve as a simple-to-use guide for those undertaking or considering automatic record linkage. All those with an interest in data sharing and matching will find this report a useful reference.

Susan Linacre
Director Methods and Quality Directorate
Office for National Statistics
January 2001
Acknowledgements

The author wishes to acknowledge the following contributions to this report:

The contributions made by the members of the Task Force on Data Sharing and Matching (see next page). In particular, I give special thanks to John Charlton, David Brown and Mohammed Yar at the Office for National Statistics for their encouragement, support and all the help they provided to prepare the final report for publication.

This report is based on the developments and experiences of the Oxford Record Linkage Study, which is a project within the University of Oxford, Unit of Health-Care Epidemiology under the directorship of Dr Michael Goldacre. The funding for the Unit was provided initially by the Nuffield Provincial Hospitals Trust (1963-1969), then by the Department of Health, NHS Executive Anglia and Oxford Regional Office and at present the NHS Executive South East Regional Office. I gratefully acknowledge contributions made by the present and past members of staff at the ORLS.

I would also like to acknowledge the international contributions made by Bill Winkler and Ned Porter of the U.S. Bureau of the Census; and Martha Fair (formerly Smith) and Maureen Carpenter of Statistics Canada.

Finally, I would like to acknowledge the burden borne by my family by sparing me time and effort during the preparation of this report.

Leicester Gill
University of Oxford
Members of the GSS(M) Task Force on Data Sharing and Matching

Bob Butcher (Chairman) Department for Education and Employment
John Charlton (Secretary) Office for National Statistics
Paul Dunn Office for National Statistics
Jennet Woolford Office for National Statistics
Dev Virdee Office for National Statistics
Catherine Ellis Office for National Statistics
Clive Jerome Office for National Statistics
Sheena Gordon Office for National Statistics
Bev Botting Office for National Statistics
Nigel Stuttard Office for National Statistics
Derek Bird Office for National Statistics
Mohammed Yar Office for National Statistics
Arthur Barnett Department for Education and Employment
Nick Kew Department for Education and Employment
Steve Kendrick National Health Service in Scotland
Sandy Stewart Scottish Executive
Trevor Benn Health and Safety Executive
Linda Shurlock Department of Health
John Bates Department of Health
Chris Morris Northern Ireland Statistical Research Agency
David Orr General Register Office for Scotland
Graham Davies Welsh Office
Phil Barnes Court Service
Edwin Ko Inland Revenue
Gillian Ferry Department of Social Security
Mike Daly Employment Service
Philip Howard Home Office
Alison Kitchener DASA, Ministry of Defence
Stephen Hall Department of the Environment, Transport and the Regions
Leicester Gill University of Oxford
Executive Summary

Record linkage techniques are used to link together data items relating to the same person, household or a business unit. The data may well be collected on different occasions and under different conditions. Linkage extends the amount of information available for analysis and enables different data sources to be combined. The statistical benefits of record linkage are immense including:

- compilation of new statistical outputs through effective use of exiting data sources
- improving data quality and integrity
- setting up longitudinal studies and profiling,
- reducing respondent burden and costs.

The first reported application of record linkage was in the area of health studies, where it was used to link patient records from hospitals and death certificates in order to study morbidity and mortality - the Oxford Record Linkage Study and numerous occupational health studies are typical examples (described in this report). More recently linkage has been making inroads in official statistics in the UK - the ONS Longitudinal Study (ONS, 1995) is a typical example, and its role is expected to increase. The ever increasing demands from users for timely, accurate and detailed statistics will oblige National Statistical Institutes all over the world to look at alternative means of producing relevant statistics, including those obtained from administrative sources. The National Statistics Framework document (HM Treasury, 2000) recognises the role of data sharing in our national statistical system.

This report concentrates on the use of record linkage for statistical purposes only, to produce summaries and statistics. Many of the administrative and survey data sets are collected under legislative framework and are subject to strict data protection and data confidentiality restrictions. The ethical and legal barriers associated with data sharing and matching are highlighted. Therefore, before attempting any record linkage, it is important to establish propriety of the record linkage exercise itself. Guidance is given on the requirements for confidentiality to protect the identity of the individuals or organisations, including the provision of a controlled, secure physical environment for the computer processing system and the data files.

The methodology for record linkage is presented in a non-technical way so that it is accessible equally to novice and experienced government statisticians. The report offers practical advice on setting up a new matching application and the resolution of the issues to be addressed. The design questions that should be considered when developing record-linkage systems for specific applications are discussed, and suitable methods presented.

There are two main methods for record linkage: the Exact Method and the Probabilistic Method. The Exact Method is suitable when a unique identifier is available for each record to be linked and the quality of the identifying data is high. The method is simple to implement and the linkage process is very fast. The Probabilistic Method is more suitable when there is no unique identifier, the data are noisy or key variables have missing values. It compares records from the two input files in pairs and works out probability-based criteria for deciding whether the two records should be linked, not-linked or possibly linked. The practical details of the method such as blocking the files, formatting the files through editing and parsing, calculating weights, and setting up thresholds are thoroughly explained.
Recently commercial software packages have become available, implementing both Exact Methods and the Probabilistic Methods. A brief description of the functionality of these packages is given. In general these packages are expensive, usually bolt onto other commercial database software and require substantial amount of training to use.

Realising the importance of record linkage for National Statistics, recommendations on how to take forward the initiative of record linkage in the National Statistics are made. The main recommendation is acquiring record linkage capability through setting up of a Resource Centre for record linkage. The Resource Centre will acquire skills and expertise, and provide technical support and advice right across the National Statistics when setting up record linkage applications. It will also carry out the necessary research and development work jointly with academics and other NSIs.
1. Introduction and aims

1.1 Introduction

The term record linkage is simply the bringing together of information from two different records that are believed to belong to the same person, family or entity. The records to be compared may come from a single data file or multiple data files. They may relate to persons or to other entities, such as households, business establishments or companies. If the two records agree on all the variables, and are unlikely to have done so by accident, the level of assurance that the link is correct will be high. Conversely, if most of the variables disagree there will be little doubt that the linkage is wrong. For intermediate situations, the methodology must predict whether the link is true or false.

During the transcription, keying and storage of the data items such as names and addresses, the introduction of errors and variation is unavoidable. And this makes the task of record linkage far more difficult. In the presence of these errors, it is not possible to be absolutely sure whether two matching records have identical values on the selected variables purely by chance or that the two records genuinely correspond to the same entity.

The data files to be matched may have been generated through administrative procedures or through surveys and censuses. A file may represent an entire defined population or only a sample from a defined population. The files to be linked may have the same or different time references. In the most general context for record linkage, input files may contain duplicates, so that linkages may have to be attempted both within and between files, and in this way deduplication can be effected.

By far the greatest use of record matching and linkage has been in health studies, where it is most frequently used for searches on large files of morbidity and mortality records. On a smaller scale, there are many and diverse uses of linkage: from the preparation of disease registers, to the provision of additional health data for an individual, and the task of purging files of duplicate entries. What is important, is that the procedures and the logic are the same throughout.

This report describes the methodology of automatic searching and record matching in a non-technical way. At the same time effort is made to make the report self contained so that it will be of use to both the novice and the more experienced record linkage professionals alike. It should be emphasised that the basic principles of record matching are deceptively simple and the technique is powerful enough to complete the task in an efficient manner, but one has to be very aware of the vagaries of real data and the manner in which these can bias the end result. These practical difficulties are clearly highlighted in the report.

Automatic record matching and linkage involves striking a balance between the efficiency of the process and the quality of the matched file. While the technology is becoming more powerful and less expensive, an insight into the mechanics of file organisation and record matching will reduce the time spent in file processing as well as providing an improvement in overall performance and match rate.

The purpose of this report is therefore to meet the needs of those who want a guide to record matching and linkage practice, and how to get started.
1.2 Aims of the report

The main aims of the report are:

1) To present the methodology of record matching and linkage in a non-technical way and this includes discussion about the data variables that can be used for record matching, the methods of cleaning up the data prior to matching, and the clerical tasks required to resolve any queries after matching;

2) To identify and discuss practical matching design questions that should be considered and addressed when developing record-linkage systems for specific applications;

3) To illustrate the potential of record linkage in official statistics through the presentation of the examples of matching exercises currently undertaken by the members of the GSS Task Force on Data Sharing and Matching;

4) To suggest some directions for further research.

1.3 Outline of the report

In Section 2, an historical perspective of record linkage is presented, in particular the evolution of probabilistic record linkage and the development of generalised record linkage systems. The uses of record linkage in official statistics, particularly in the area of health statistics, are illustrated through presenting examples of the well known applications of record linkage.

The two commonly used methods for record matching are the exact method (or deterministic method) for matching high quality data, and the probabilistic method for matching low quality or noisy data. These methods are introduced in Section 3 and described in more detail in Sections 6 and 7 respectively.

For those new to record linkage, a road map to setting up a new record linkage application is presented in Section 4. The three stages in the linkage process are presented in this Section, together with the choices on offer and the pitfalls for the unwary. The different approaches to the choice of matching method using parameters like file size, and number of matching variables are also presented.

In Section 5, the current practices in the data capture and subsequent cleaning processes referred to as data editing and parsing are described, together with the way in which the records are blocked and stored in the computer system. The use of phonetic compression codes for the blocking of the data and master files is described. Also discussed are the types of problems to be overcome where there are errors or omissions in the blocking keys.

For simplicity and speed the exact match is to be preferred where possible. This type of match is presented in Section 6. It relies on an universally available identifier, is easy to implement, quick to process and the result is either a match or a non-match. If such a single identifier is not readily available or is unsuitable, a partial identifier can be used singly or in combination with other partial identifiers to effect the matching.

Where a single identifier is not available for matching, or the data contains errors and omissions, the match can still proceed using a group of partial identifiers and a probabilistic matching method will need to be used. In Section 7, the theory and application of the probabilistic match are presented, together with the Fellegi-Sunter theory (see Annex A) that underpins the methodology. The problems inherent in selecting the match acceptance thresholds used for the probabilistic match are presented in Section 7. In Section 8, the
methods of building a file of linked records are discussed and also the problems that arise when there are logical conflicts in the construction of the file.

The results from the selected on-going and planned GSS projects on data sharing and matching are described in Section 10, the full details of which are included in Annex G. These projects were selected to provide the widest range of record matching activities to illustrate how universally the methodology can be applied.

In Section 11, the future developments in record matching and linkage are presented. While the methodology is now well established, the method of coping with the query outputs and the manner in which they are evaluated in large systems need further development. To make any significant improvement in the art of matching, advances in file blocking and searching are the key elements to be improved. Estimates of the match rate and more importantly the false positive rates (records matched to the wrong person) will need to be measured, this being especially true for the larger files.

The data protection aspects of the matching process are presented in Section 12. This Section does not attempt to cover the ethical issues associated with record matching, but the data protection and security required for the processing of computer files that contain real identifiers.

The techniques of data sharing and matching offers numerous potential benefits to National Statistics including improving the quality of existing outputs, resulting in new outputs and at the same time reducing respondent burden. In Section 14, recommendations on how best to take forward the initiative of data sharing and matching in National Statistics are made.
2. Development of record matching and linkage

In this Section the development of record matching and linkage over the last forty years is presented. The main focus is on medical record matching and linkage, the manner in which it started and how it has developed.

2.1 What is record matching and linkage?

The basic concepts of record linkage should be familiar to all of us, since we apply them whenever we look for a number in a telephone directory, a service in the yellow pages or a product in a catalogue. We start with certain information, it could be a name (although we may be uncertain about the spelling), a description, a town, and, possibly, a street and number. The scope of our search is limited by the blocking and sorting techniques used in the compilation of the directory. To derive a telephone number we examine the directory for the appropriate geographic area and, using the latest UK directories, select the section for individuals or for business and professional organisations. We then search for the name in the alphabetical index since the telephone number for individuals is blocked and sorted on surnames followed by their street address. To cater for the various spelling variations in the subscriber’s names, the directory uses a 'see-also' approach to draw attention to the various spellings for the common surnames, for example STEWART and STUART.

Where we cannot find a unique entry which is in full agreement with all of our matching criteria, i.e. surname, initial and street address, we look for entries that are in partial agreement and we place some of these in the possible match category. In this category we make implicit judgements about how much weight to give to the names in the directory for each of the matching variables. These possible links are then resolved by telephoning the subscribers, starting with the one deemed most promising, until a positive link is established.

While the basic ideas used in record linkage are a simple extension of the above example, there are many interesting and challenging technical problems that must be solved in undertaking large-scale record linkage. The record matching and linkage theory and models developed by Fellegi and Sunter (1969) and others, offer valuable guides to solving these problems.

2.2 History of record matching and linkage

Although there are a few early references to the desirability of record matching and linkage (Farr, 1861; Nightingale, 1890; Stocks, 1944), the main thrust in this research area came in the 1950’s with the ability to store medical records on a computer. The typical matching process, then and now, involves the merging of two medical records, taken at different times and in different places, so that the new information is correctly linked to the earlier medical record for the same person.

2.2.1 Early developments in record matching

Prior to the advent of computers, most record linkages were carried out using manual procedures based on ad hoc rules and decisions. The matching was assigned to clerks who reviewed the printed lists, obtained additional information when matching information was missing or contradictory, and made linkage decisions following empirically established rules. Typically, these files were sorted alphabetically by name or address to simplify the review process. If a name contained an unusual structure or spelling error, the clerks might not find its matches. For large files, matches could be separated by several pages of printout, so that some matches might be missed. Even after extensive training, the clerk’s matching decisions
were not always consistent. Human judgement played a major role in deciding which records to match together, and the borderline cases were often resolved, not by recourse to the set of explicit rules, but by giving the cases to one or more 'experts' deemed best qualified to judge whether or not the two records belonged to the same person or entity. Where two records have sufficient information to make decisions about whether the records represent the same person or entity, humans can exhibit considerable ingenuity in recognising unusual typographical errors and abbreviations, and making allowances for missing data.

Compared to manual matching, computer matching has the advantages of allowing central supervision of processing, better quality control, higher speed, and more consistency and reproducibility in the results. For all but the most difficult situations, modern automatic record linkage can achieve results at least as good as that using highly trained clerks. For those cases where additional information cannot be merged to the files automatically, humans are often superior to computer matching algorithms because they can cope with a variety of inconsistent situations.

2.2.2 Advent of computing systems for information processing

The key technological development was the shift from a paper based systems of record keeping to a computerised process for the creation, transmission and distribution of the information. Computer software suites have been developed for the repetitive processes of sampling, data capture, automated coding, editing and imputation. These suites of software products use technologies that make them portable across the main computing platforms, from mainframe computing systems to desktop personal computing systems.

As computers became widely available in the 1950s, it was natural to try to exploit their power and reliability for large-scale record matching activities. As with other applications of computers, this required careful and constant review and rethinking of the manual and ad hoc procedures, and the development of the explicit specifications for every step in the process.

The first definition of record linkage was in terms of the book of life:

... Each person in the world creates a book of life. The book starts with birth and ends with death. Its pages are made up of all the principal events in life. Record linkage is the name given to the process of assembling the pages of this book into one volume. The person retains the same identity throughout the book. Except for advancing age, he is the same person. ...

Source: (Dunn, 1946)

The early developmental work investigated the feasibility of applying probabilistic matching and record linkage (Newcombe et al, 1959; Acheson et al, 1963) for the collation of existing morbidity and mortality records. The use of frequency ratios (essentially statistical likelihood ratios) to quantify the degree of matching was established empirically. The main theoretical support for automatic record matching methods was firmly established by the late 1960's with the papers of Nathan (1967), Tepping (1968), D'Andrea Du Bois (1969) and Fellegi and Sunter (1969). The practice of record linkage dates back even earlier, at least to the 1950's although the work of Newcombe and his collaborators are regarded as seminal to the science of record linkage. (see for example, Newcombe et al, (1962); Newcombe, (1967)).

Despite these promising early developments, most record linkage activities in the 1970s and the 1980s continued to be performed using ad hoc heuristic methods. There were a number of reasons for this as follows:
1) There are only a few people whose main professional interest is record matching and linkage. As a consequence much of the applied work in this field has been done by individuals who may be solving matching problems for the first time and empirically deciding which combinations of matched variables constitute matched records. Because the basic principles of record matching appear to be simple (but in fact are extremely complex), some of the ad hoc solutions that have been used were far from optimal. It is relatively straightforward to develop ad hoc record linkage methods that may achieve matching rates of between 50–80 per cent. This is completely different from other areas such as sampling, estimation, editing and imputation where a moderate knowledge of the subject is always helpful.

2) Statisticians typically get involved very late in the matching stage, often after the files have already been matched and the new files created. Even when this is not the case, little emphasis may be placed on the data structures needed for linkage, because other uses of the data are usually assigned higher priorities. Specific design opportunities have, therefore, generally been limited, since the files were produced largely for other purposes.

3) Until the 1980s good, portable, general-purpose matching software was not commercially developed and distributed widely (Howe et al., 1981), despite some important early attempts (e.g. Jaro, 1972; SSA, 2000; Hill, 1990). Even in the presence of general purpose software, the uniqueness of each matching environment may require practitioners to prepare complex customised programs both for data formatting and for record matching, and in the process may absorb resources that might have been better spent in other ways.

4) Particularly in respect of matches to administrative records, barriers to the introduction of improved methods have existed because cruder methods were thought to be more than adequate for some administrative purposes.

The analysis of linked data sets is still in its infancy. Researchers have written extensively about the limitations of record matching when applied to their applications. This is in contrast to descriptions of the expected success rate and the results that have or could be been achieved.

The key technical issues were identified fairly early in the development of computerised record matching and linkage (Smith, 1984) and fall into three main areas which are concerned with the use of personal identifiers, errors in reporting and the size of the file. These are presented in Box 2.1.

2.2.3 Development of probabilistic methods of record matching

The earliest matching processes used the exact match to search the large databases quickly for records on the master file that matched exactly with the records on the data file. There was no automated way to compare the names other than to identify that they were exactly the same or not. Where the two name strings agreed exactly, the records were considered to be matched together, otherwise they were regarded as belonging to two different people. Exact match systems are still used today because of their simplicity and high speed but they have the drawback that no approximate matches are returned.
Box 2.1 Key technical issues in the development of record matching

1. Using personal identifiers to discriminate between the person to whom the record refers and all the other persons in the population.

2. Deciding whether discrepancies in identifiers are due to mistakes in reporting for a single individual or to the presence of other individuals.

3. Processing the large volume of data necessary for record linkage within a reasonable amount of computer processing time.

Source: Smith, 1984

Person names are not normally unique, except where they are combinations of very rare surnames and forenames. Names when spoken, written or captured in a computer system are subject to considerable variation in recording and spelling. Even if the names stored in the computing system are accurate, the name from the data file will come from the real world and be subject to error and variation. Non-name data such as dates of birth, address and postcode are all subject to error, truncation and incompleteness. Complications arise when the identifying set contains: spelling errors, data preparation errors, use of synonyms and nicknames, anglicisation of foreign names, initials, truncation and abbreviation, missing words, and extra words. Due to the fact that certain names like SMITH are 10,000 more frequent than SZABO, there will be many more surname/forename combinations to be searched. The use of an exact matching method (see Section 6.1) would fail to bring together records which contain even small amounts of error, omission or over qualification in any one part of the identifying set. For these reasons the non-exact or probabilistic methods of record matching have been developed.

2.2.4 Introduction of phonetic codes into record matching

A person who is aware of the ways in which surnames can be spelled, can search the master files under the various versions of the surname using some see-also algorithm such as that incorporated in a telephone directory. To avoid the use of such tables of synonyms in computerised searches, phonetic coding schemes have been developed for blocking and searching the files. To provide some method of coping with the various spelling errors on records that are captured from the various feeder systems, the Soundex phonetic algorithm was developed to equivalence the different spellings of the surname. The exact matching approach could then be used to bring together records that had the same Soundex code although the recorded names could be spelt differently.

The problems inherent in the original Soundex scheme inspired the development of other phonetic compression methods, which in 1963 resulted in the development of the New York State Identification and Intelligence System (NYSIIS). Nearly all advanced phonetic compression algorithms are variations of this refinement of Soundex. The benefits of the NYSIIS system include the coverage of more spelling variation, especially with ethnic names, and fewer coded names are returned than with the Soundex algorithm. However the underlying one-algorithm-fits-all processing of these new systems remains the same and only masks the fundamental weaknesses in the Soundex approach. Names are very complex structures, and the NYSIIS algorithm still misses many close matches and there is little sensitivity to cultural names. (see Annex B)

The more recently developed name searching algorithms start by first determining the cultural origins of the name. This information, together with the appropriate computational techniques
can then be used to find the best name matches within the target culture, for example, unlike earlier attempts at general name searching and matching, different techniques can be developed to search for Chinese name matches, or for Arabic or Indian Sub-continent names. Recent work by the ORLS on the identification of Asian names in the UK has been used to prepare lexicons of Asian surnames and forenames, and is used for the calculation of the appropriate outcome specific weights (see Section 7.3). There was a need to prepare a lexicon of Arabic names, in fact there was even a need to prepare a lexicon of the specialised surname and place name spellings used in Wales.

2.2.5 Development of general purpose matching and linking software

The first matching programs were prepared in the early 1960's by Government and University departments using the software tools available to them at that time. The software was mainly concerned with the comparison of two records which were held in character format and computing a measure which reflected the amount of agreement of disagreement between the records and in this way effect a link.

The U.S. Bureau of the Census began work on a generalised record linkage system, designed primarily to evaluate census coverage by linking census and administrative records (Jaro, 1985). Two U.S. agencies that conduct economic surveys, the Statistical Reporting Service of the Department of Agriculture (Coulter, 1985) and the Energy Information Administration (Winkler, 1985a) have developed record-linkage systems for use in constructing sampling frames from multiple list sources. In 1980, the National Centre for Health Statistics established the National Death Index (NDI), which contains computerised records for all deaths occurring in the United States from 1979 onwards. Health and medical researchers use the NDI to determine which persons in their study populations have died. Research and operating experience has led gradually to improvements in the record-linkage procedures used in the NDI operations (Patterson and Bilgrad, 1985). The ORLS, started in 1963, have developed suites of record linkage programs which have been used successfully to prepare linked files which span 1963 to 2000.

The 1980s brought a resurgence of interest in the development of sophisticated general purpose computerised systems for record matching and linkage, based on models similar to those proposed by the pioneers of the 1950s and 1960s. The software developments in the public and academic sectors include: the Generalised Iterative Record Linkage System (GRLS3) developed by Statistics Canada, the National Cancer Institute of Canada (Smith and Silins, 1981), the California Automated Mortality Linkage System (CAMLIS) developed by the State of California (Arellano, 1985), OXLINK developed by the University of Oxford (Gill et al, 1993, 1997), record linkage in Scotland (Kendrick et al, 1997), and record linkage in Western Australia (Holman et al, 1999). In parallel, generalised record linkage software was developed commercially and included SSA Name3, Intelligent Retrieval, Trillium, Integrity, Quick Address and PA Oyster Engine. For details of a selection of software packages refer to the Section 9.

2.3 Record linkage applications in the UK

A number of large record linked datasets are in operation in the UK, and include the following well-known and documented systems:

a) The Oxford Record Linkage Study (ORLS)

One of the pioneering practical studies of record matching in the UK health field was undertaken by the Oxford Record Linkage Study (ORLS), under the directorship of Dr Donald Acheson (1967). The ORLS successfully adapted the methodologies developed at
Statistics Canada by Howard Newcombe for use with the UK health and vital records. The ORLS was started in 1962 with financial support from the Nuffield Hospitals Provincial Trust and subsequently by the Department of Health. The ORLS was set four tasks, namely to:

1) study the feasibility and cost of prospectively accumulating information about key health events, for a defined population, in cumulative personal and family files,

2) develop computer methods for record matching and linkage,

3) study applications of linked files in medical and operational research,

4) promote its extension on a national basis.

The initial data were limited to brief extracts of each birth, hospital inpatient discharge and death for a population of about 350,000, and it was hoped to link these data together using the National Health Service (NHS) number. In practice only a minority of the records contained the NHS number and the decision was made to adopt and use the probabilistic method being developed and used in Canada (Newcombe, Kennedy, Axford and James, 1959). The study was extended to include all of the Oxford health region by 1985 with a population of 2.5 million. The applications using the ORLS file include the statistical analysis of person-based longitudinal files, and tables of hospital morbidity rates for a range of conditions. Later developments include: studies of the association between diseases, outcomes, and studies of the health services.

This study outlined an important goal of comprehensively matched and linked population databases, that is to have:

... a system of linked health records which brings together selected data of biological interest for a whole population commencing with conception and ending with death, in a series of personal cumulative files, the files being organised so that they can be assembled into family groups. The term record linkage may apply specifically to the techniques of assembling the files in spite of errors and omissions in the identifying particulars, or may be used in a more general sense to apply to the organisation involved...

Source: (Acheson, 1967)

This goal has been restated many times since 1967. The following statement by the then Chief Executive of the UK NHS, Duncan Nichol, covers the concept of seamless medical care:

... If we are to have seamless care and the ability to track individuals, whether in hospitals, or between the health care providers, then the Management Executive recognises that we must have, throughout the NHS, a unique person identifier. Even within one organisation, such as a hospital, we are likely to find several identifiers for any one individual let alone between organisations. This is a real barrier to integration. Moreover the absence of an identifier used in all parts of the NHS means that we are having to depend on name link processes. That makes the task of preserving confidentiality much more difficult...

(Duncan Nichol, CEO, NHS Executive, 1992)
b) The ONS (formerly the Office of Population Censuses and Surveys) Longitudinal Study

The ONS Longitudinal Study, which traces its origins to the Oxford study, is one of the most important developments which has taken place in record linkage in Britain since 1962. It is a significant development in social and medical statistics.

The study started in 1974 with a one per cent sample drawn from the resident population of England and Wales enumerated at the 1971 census and containing Census and vital events. The sample was drawn by selecting everyone born on any of the four particular days of the year who were in the 1971 census. Subsequent samples have been drawn and linked from the 1981 and 1991 censuses. The change in the population is reflected by the addition of new sample members born on one of the four dates, together with people exiting from the file through death or emigration. Linkage of the data became possible with the recording of date of birth rather than age in the decennial census and at birth and death registration. The matching of the file is undertaken by the National Health Service Central Register (NHSCR) using data from subsequent censuses, the national cancer registry and death certificates. The records of persons born on the specified dates after the 1971 census are added to the file, together with records of immigrants born on these dates. The longitudinal study contains selected records arranged in personal cumulative files.

The uses of the longitudinal study include the analysis of occupational mortality, and to provide better information on fertility and birth spacing. Further uses include the analysis of migration and other socio-demographic studies (Hattersley and Creeser, 1995; SSRC, 1998)

c) Record linkage in Scotland

Heasman and Clarke (1968, 1979) published a description of the medical record linkage which existed on a national scale in Scotland. In addition to all Scottish births and deaths, hospital in-patient, and cancer registration records, the system also included child health records. Initially the linkage was performed on an ad hoc basis and up to 60 linkage projects were carried out (Kendrick, Clarke, 1993). Development of the current system began in May 1989 with the long-term aim that all the records held centrally should be linked together.

The system is being used for the analysis of episodes, stays and patients and the calculation and analysis of readmission rates. Other studies based on the linked file include analysis of mortality, and the modelling of outcomes. In parallel with the other research units presented above the emphasis has shifted from the development of record matching and linking software to improving the methods of analysing, presenting and applying the results of the linkages.

d) Cancer registries and other national screening and follow-up systems

The framework for a national follow-up system predated the ORLS, although it had not been used for this purpose. It consisted of the NHSCR which, since 1948, has contained brief details of every person registered with a general practitioner, updated for deaths and emigration. During the 1970s, information about cancer cases from the National Cancer Register were added to the NHSCR. The system of record matching was a manual one based on clerical comparisons of the identifying particulars. With the conversion of the NHSCR manual index to a computer database in 1990, the matching is now undertaken automatically by the computing system using a exact method and backed up by clerical checking.

Other person linked projects include the creation of child health files, screening files such as those of cervical cytology and breast screening, and employment and turnover data from the Inter-Departmental Business Register (ONS) and floospace data.
Record Linkage is also an important tool for the creation of statistical data, particularly in relation to large-scale data collection activities and to census issues. Some of the important uses of matching and record linkage in healthcare are presented in Box 2.2 and other uses of National datasets are presented in Box 2.3.

2.4 Uses of record matching and linkage

Record linkage has also been used in a wide variety of analytical studies, but in general it is tied to the increased use of previously collected and coded, clinical and administrative records for statistical purposes, and the reduction in the burden and cost of data collection. Detailed description of some of the uses of record linkage have been described elsewhere (Fair, 1995; Newcombe, 1994; Goldacre, 1983, 1986, 1993), and some typical applications are presented here.

a) Data quality and data cleaning

Record linkage is used to prepare a file of linked person or entity records and this can lead to improvements in data quality through logical checking of the records and borrowing or propagating various data items over the whole sequences of records for a person. Some European countries use localised population registers in addition to the census and it is possible to match administrative data and use record linkage to help impute missing or inconsistent data. Data sources can then be processed to eliminate duplicate records for individuals and to identify missing records in databases (e.g. by the linkage of infant deaths and birth records or by the linkage of births and deaths with census records). In the UK, US and Canada, record matching the data is used to improve the census coverage (e.g. address register) as well as to estimate its coverage (e.g. reverse record check). Combining the results from several record matching exercises using various data collection sources may give improved estimates of population or expenditure (e.g. use of income from tax, survey and census sources).

b) Preparation of disease specific registers

In the preparation and use of disease-specific registries, record matching can be used to identify under-reporting of cases, e.g. by linkage of cancer registries with death registrations, or the linkage of hospital records. Record matching has been used with some success for creating cancer registry data from hospital events and vital records. Some cancer registries combine a variety of data sources and use record matching to generate their registry from hospital admissions, pathology reports, records from clinics, and death registrations.
Box 2.2 Applications of Record Linkage in Health Studies

Use of record matching in Health Services research and operation of the NHS
Hospital Enquiry Statistics (HES)
  use of DOB/Sex/Postcode as a key for exact matching
National Health Service Central Register (NHSCR)
  tracing and tagging
Exeter File (Mirror site for NHSCR)
  Migration, de-duplication, detection of errors in the update processes
DHA files
National Strategic Tracing Service (NSTS)
Strategic roll out of the new NHS numbers

True counts of people as opposed to counts of episodes of care
  True counts of patients as opposed to counts of episodes of care on HES
  Person spells vs. activity spells
  Studies of re-admission rates to hospital

Seamless medical care
  Creation of patient-oriented histories (longitudinal files that cover all events)
  Generation of the Electronic Patient Record (EPR)

Matching with Primary care services
  referral rates
  prescribing rates
  use of laboratory and other services

Screening and follow-up
  Support for population screening procedures
  Breast cancer screening
  Child health systems
  Follow-up surveys (e.g. Nutrition Canada, Canada Health Survey, Fitness Canada)

Vital records and registration processes
  Matching deaths to morbidity records
  Death tagging and clearance
  Matching death records to birth records for children under 1 year old

Issue of National Numbers
  Issue of national numbers (e.g. new NHS number) and checking accuracy of issue

Use of record matching for Epidemiological and Health-Care studies
  Identification of cohorts: specific diseases, surgical operations, ethnic groups,
    age/sex groups
  Matching index cases with controls
  Outcome and follow-up measures
  Hospital morbidity in early life
  Disease association studies
  Regional variations in the incidence of disease
  Studies of survival following a diagnosis of cancer or progressive disease
Box 2.2 (continued) Applications of Record Linkage in Health Studies

Examining factors which influence health care
  Mortality, cancer and/or birth follow-up of,
  Cohorts (e.g. wood workers, miners, asbestos workers)
  (Case-control) studies
  Longitudinal studies
  Clinical trials (e.g. Canadian Breast Screening study)

Preparing and maintaining disease registers
  Building, maintaining and using registries (e.g. cancer and AIDS)

Studies of occupational disease
  Occupational and environmental health studies

Pharmacology
  Drug monitoring
  Post-marketing surveillance in pharmaco-epidemiological studies

ONS Longitude Study
  Long term follow-up of the 1971 cohort

Genetic studies
  DNA sequencing:
  Follow-up of special phenotypes

Creation, de-duplication and cleaning of index files
  Creation of comprehensive multi-file databases
  Reconciliation of two master files
  De-duplication of patient master files

Other applications
  Costings and their accumulation
  Forecasting

Source: Smith (1986); Goldacre (1986a, 1986b, 1987); Gill and Baldwin (1987)

Box 2.3 Some Applications of Record Linkage to other types of National Datasets

Office for National Statistics (ONS)
  NHSCR
    allocation of the new NHS number
    de-duplication
    automatic matching to the register
  Vital statistics
  National cancer registry
    flagging of deaths for occupational and rare diseases studies

DfEE
  Schools
    Exam results
    employment
    higher education

DETR (Department of Environment, Transport and the Regions)
  Retail Planning
    Employment and turnover
    Office floor space
    Geographic information system
c) **Patient Oriented Records or the Electronic Patient Record (EPR)**

Creation of patient-oriented, rather than event-oriented statistics (e.g. for hospital admissions, for cancer registries, (Dale, 1989). Such systems may be used as a sampling frame for health studies (e.g. matching of mothers with their female children for examination of possible psychological problems passed from one generation to the next).

d) **Tracing Tool**

Record linkage and administrative records are often used to follow-up cohorts to determine the individual's vital status. Tracing is often needed for follow-up of disease-specific cohorts and for longitudinal surveys to obtain the cause of death or cancer. Mobility patterns of persons are important for the allocation of health resources.

e) **Supplementary longitudinal surveys and screening programmes**

Several surveys have been carried out following certain groups within the UK. Examples include matching and identifying specified cohorts and the health and activity needs of this group. Data from these surveys could be linked with that available from health authorities or the NHSCR.

f) **Construction of sampling frames**

The construction of the ONS Longitudinal Study is a good example of a sampling frame that has been assembled using record matching and linkage techniques. The original cohort drawn from a census, is augmented each year by new persons who fit the criteria for membership. However these surveys are susceptible to attrition bias, which results in loss of members from time to time. This loss can be due to death, or through change of name or address. This can lead to the sample being unrepresentative since it will only represent the characteristics of those members that still remain. Additional records for members of the cohort can be added to the file and matched in with the existing records. This file can then be sampled in a number of ways, both linked and unlinked.

g) **Genealogy and history**

Large-scale record linkage is being undertaken by the Mormon Church at their International Headquarters in Salt Lake City for the linking of families. Linking records has always been fundamental to the process of historical enquiry, and since the 1960s the rules developed for medical record linkage have been applied to historical data for tracking family groups over many generations.
3. An introduction to record matching methods

This Section focuses on methods for evaluating potential matches between records on a data file and records on a master file. Many techniques have been developed for record matching, some of which are very basic and rely on the exact comparison of the identifying set, while others are more sophisticated and mimic the actions of a very experienced coding clerk.

Record linkage involves three stages. The first stage involves bringing potential matches together so that they may be compared. The second stage involves the comparison of the potential record pairs to decide whether they belong to the same person/entity. This could use an exact (deterministic) matching method using a universally available high quality identifier, for example an assigned number like the NHS number, as described in Section 3.1. Alternatively a combination of the various partial identifiers, for example, names and addresses, can be used to compute scores called weights for each potential match based on probabilities (probabilistic matching). The resulting weight is then compared with a preset threshold determined from a theoretical approach, or from empirical results computed from previous matches, to decide whether to mark this pair as a truly matching pair. This is described in Section 3.2. With the advent of the World Wide Web and on-line access to large databases, other matching methods have been developed, and these are described in Annex C. The third stage involves the collation of the matched records into the matched file, as described in Section 8.

In the following discussions, assume that the records consist only of formatted text. Record linkage involves bringing together records on the data file and potentially matching records on the master file, and assessing whether there is enough evidence to assert a true linkage between the two records. The record on the data file is usually called the query or data record and all the potentially matching records on the master file are called the candidate or master file records. Each record is composed of variables or identifiers which contain the information on which the match will be evaluated. The identifying variables on the data file must have the equivalent variables on the master file, for example, surname or company registration number. To use a computer for the task of matching a data record with records from the master file, the process must be reduced to an evaluation of each record pair on the basis of some computable function, or measure. The differences between the methods hinge largely on how this measure is calculated for each pair, and how the measure is evaluated to determine whether the pair should be regarded as a true match or not.

3.1 Exact matching (deterministic, all-or-none methods)

The main requirement for exact matching (sometimes called deterministic or all-or-none matching) depends upon the records on both files containing a variable or characteristic of a person or object that is ideally (i) universally available, i.e. available for all records, (ii) fixed, (iii) easily recorded, (iv) unique to that individual, and (v) readily verifiable. Few, if any, variables meet all these requirements, though several come sufficiently close to be usable. The perfect variable would be an integral feature of the individual, such as a personal trait, or a group of traits which together form a set that is unique to that individual. Alternatively, unique identification numbers may be assigned to individuals at birth, or a unique number or cipher allocated to the person or object using a highly reliable and accurate procedure, for example national insurance numbers, bank account or other national or catalogue numbers.

Systems of numbers or other ciphers can be generated which meet these criteria within a given setting. In the healthcare environment (e.g. within a hospital or health district), the roll-out of the new ten digit NHS number has improved the prospects of exact linkage and the new NHS number will be incorporated in all health care records from 1997 (Secretaries of State, 1989; National Health Service and Department of Health, 1990).
Exact matching generates links that are based on the agreement of the selected identifying variables on the two records. In the simplest version of exact record matching, the output of the match is clear cut: either the records match or they do not. This tactic simplifies the record linkage methodology, making it more practical for quick jobs and for rapid processing on small computing systems (see Section 6). For comparison of records which contain many variables in which there is a possibility of a low error rate, there is another version of the exact matching, which relaxes the exact match criterion a little. Then the number of variables that agree are used to determine whether the record pair should be linked: an almost-exact match is all that is required. This is different from probabilistic matching (see below), since it utilises a very simple matching criterion and does not require probability weights.

3.2 Probabilistic matching

Probabilistic matching methods have been developed for data and master files that contain errors and omissions, and for which there is no unique or universally available high quality identifier. We no longer require an exact match, and partial matches are assessed with more sophistication than described at the end of Section 3.1. If all the selected variables on the data and master file record agree, and are unlikely to have done so by accident, the level of assurance that the records belong to the same person will be high. Conversely, if they all disagree and are very unlikely to have done so in truly linked pairs of records, there will be little doubt that the records in the pair are wrongly matched. For intermediate cases, the evidence must be evaluated to decide whether the records possibly match. The critical features of probabilistic matching involve the calculations which quantify the two italicised phrases in the above sentences. How this is done is discussed in principle below, and in mechanistic detail in Section 7.

Probabilistic record matching is so called because it relies on calculating scores based on probabilities. The method involves determining agreements between variables in the identifying set in the two records, and also disagreements. One could use a weight based on the number of agreements minus the number of disagreements as a basis for the decision whether the record pair should be regarded as truly linked or not. This is more or less what is achieved in exact matching. Probabilistic record matching carries the calculation a little further. It involves asking the questions mentioned in the introduction to the section. Either from previous experience of record matching in similar areas of application, or based on a preliminary matching exercise carried out on the current data, how likely is it that the variables which matched in the current record pair would have done so by chance if the records were not correctly linked? This is compared with how likely the agreement would be in correctly linked record pairs. Clearly what we need to use in any reliable record matching procedure are those agreements between variables which are more typical of correctly linked pairs, rather than those which might well have occurred by chance in unrelated records. The features that might agree by chance in unlinked record pairs are those which don't divide the population under consideration into many subclasses, e.g. gender or marital status (married, single, divorced, widowed). A characteristic that is much more useful for this purpose is the date of birth. Birthday alone (ignoring the year) divides the population into 365 subsections, and is therefore much more useful as a component of a matching variable, although clearly it is not sufficient on its own, because on average $1/365^{th}$ of the population will have the same birthday.

In its standard form, probabilistic matching therefore requires some preliminary matching to have been carried out, or to use experience from a previous study. Using these data, for each variable in the matching identifier set that agrees with the corresponding variable in the record pair under consideration, the probability that it would agree in truly linked records is calculated. This is then compared to the probability that it would agree by chance, or as a result of coding error, in unlinked records. Scores based on ratios of these probabilities (or
relative frequencies) are calculated. These scores are called frequency ratios, after Newcombe et al (1959):

\[
\text{FREQUENCY RATIO} = \frac{\text{relative frequency of agreement (x,y) among linked pairs}}{\text{relative frequency of agreement (x,y) among unlinked pairs}}
\]

(for agreement) \hspace{1cm} \text{relative frequency of agreement (x,y) among unlinked pairs}

and similarly for disagreements,

\[
\text{FREQUENCY RATIO} = \frac{\text{relative frequency of disagreement (x,y) among linked pairs}}{\text{relative frequency of disagreement (x,y) among unlinked pairs}}
\]

(for disagreement) \hspace{1cm} \text{relative frequency of disagreement (x,y) among unlinked pairs}

where:

x indicates the value of the identifier on the query or data record, and
y indicates the value of the identifier on the candidate or master file record.

These scores are transformed (by taking logarithms) to give what are termed weights, and then summed over all the identifying variables for the record pair to give an overall score for the record pair. There will almost always be a plus sign for weights for fields which agree, and a minus sign for those which disagree between the pair of records under consideration, since non-linkage is usually more likely when the variables disagree, and linkage more likely when they agree. This summed weight is then compared with some threshold value, determined a priori, to decide whether the overall weight for this record pair is high enough to classify the pair as a true match. Alternatively, whether it is low enough to classify it as a definite non-match; or if it is between these thresholds, then to either classify it as a query match, or to put it into a category which will be examined further, possibly by an experienced clerk. More detail about how weights are calculated from the probabilities or relative frequencies is given in Section 7.

There is, however, one important complication which is worth describing here. Newcombe et al. (1959) introduced methods for using the specific values that the matching variables take in the calculations of the weights. To take a simple example using forenames, a pair of records containing the forename LEICESTER will have a higher probability of belonging to the same person than a pair of records containing the forename JOHN, since the frequency of LEICESTER is about 1/3000 of the frequency of JOHN. In this example the forename LEICESTER has more discriminating power than JOHN. So the weight is different for each possible outcome of a specified identifying variable. For this reason, the term that is used to describe this complication is outcome-specific weights.

For probabilistic record matching to be cost effective and efficient, it is necessary to block together records that are likely to refer to the same person, thereby reducing the time spent searching the file, and only making comparisons within these blocks (see Section 5.3). Clearly, the reliability and efficiency of the matching procedure is highly dependent upon the way in which the initial blocking is carried out. One important consideration is that the number of records in each block is small enough to avoid too many unproductive comparisons and yet large enough to prevent records for the same person spilling over into adjacent blocks and so failing to be compared. The balance between the number and size of the blocks is particularly important when matching large files. The selection of the variables used for the file blocking is, therefore, crucial to the overall performance of the system and the efficiency and accuracy of the matching process.
3.3 Other record matching algorithms used for record linkage and text retrieval

Many other record searching and matching techniques have been developed for the matching and retrieval of textual information. Many rule based searching and retrieval algorithms have been developed over the past ten years in support of the search engines supporting access to the World Wide Web. The list given in Annex C, while non-exhaustive, is a guide to the typical methods being developed.
4. Practical procedure for record matching and linkage

In this Section the focus will be on how to implement record linkage on real data sets using either an exact matching method or a probabilistic matching method or a combination of the two. Whether exact or probabilistic matching is used, there are three main stages of the matching process. The first stage is the data cleaning stage, known as the pre-processing stage. It is the most labour intensive of the three and is described in detail in Section 5. The second stage is the choice and implementation of the matching method and it is a relatively short step when using modern computing systems, even in a multiple pass system. Usually the choice is between exact matching and probabilistic matching, and these methods are described fully in Sections 6 and 7 respectively. The final stage is ‘building the linked file’ where the uncertain matches need to be resolved usually manually and it is described in detail in Section 8. The logical sequence of these operations is shown in Figure 4.1.

Over 75 per cent of the effort for record matching and linking two files together is expended in cleaning and parsing the two input files (see Section 5). The variables judged to be the best for file blocking are given additional editing and checking since the success of the matching process is highly dependent upon these variables being as accurate as possible. The next five per cent represents the matching and linking efforts and the remaining 20 per cent represents the efforts of clerically checking that the computer matching is correct.

4.1 The pre-matching processes

Independent of the method of matching used, it is far more efficient to undertake record matching where the input and master files are converted to the same standard format. The major element of the pre-matching process is editing of the matching variables for errors and omissions, and for range checking the numerical variables. This process can be long and tedious, but the overall result of the matching process is highly dependent on getting this step as accurate as possible. It usually involves cleaning the data files, selecting the matching variables and blocking and sorting the files. These and the other elements of the pre-matching process are depicted in Box 4.1 and briefly described in the following:

Select the matching keys and other matching variables

The variables used for record matching and blocking the file need to be carefully selected, and this process is described fully in Section 5.1. If no unique identifier(s) is available, an identifier may be defined which consists of a combination of partial identifiers. If the exact matching method is selected for the record matching process, the simple key or the compound key needs to be as complete as possible since only records with absolutely identical keys will be blocked and matched together. In the case of a compound key, there is the possibility of a collision since a number of records could have the same key. For example, where the key is a combination of date of birth/gender/postcode, identical keys could be generated from the records of similar sexed twins. Where collisions are expected to occur then an extra key needs to be used and could be based on birth order, forename or initial, or some other identifying variable, to uniquely identify and separate the people who have identical keys (see Section 5.1).
Box 4.1 Stages for record matching and linking

4.1 The pre-matching processes
Select the matching keys and other matching variables
Clean up and convert the files to the same or similar formats
Apply any editing and standardisation to the variables
Add the record headers for blocking and sorting
Add any other derived variables
Select the manner in which the files will be input to the matching software
Block and sort the files

4.2 Select the matching method
Universally available and stable variable(s) for identification, use exact matching
Noisy data, use a probabilistic method

4.3 Match the file
Set and apply the acceptance match thresholds
Clerical intervention
Apply any corrections

4.4 Link the matched records to the rest of the file
Logical editing
Clerical intervention
Apply any corrections

4.5 Build the linked file and check for errors
Test for Type I and Type II errors

Clean up and convert the files to the same or similar format.

Record matching two files together is more efficient if the format of the data files and the master files are identical, and the corresponding variables have the same characteristics, i.e. field length and coding status. It is normal practice to add a header portion to each record which contains the blocking keys, although this is largely determined by the requirements of the sorting package or method of data encryption. Each record will need an identification number in addition to a reference to the person to whom the record belongs (see Section 5.2).
Apply any editing and standardisation to the variables.

Before any attempt is made to record match the data and master files, the variables must be standardised. Firstly, the variables that have been selected for blocking and record matching should be rigorously edited and any names and address fields should be parsed (see Section
5.2). The accuracy of the match is dependent upon the quality of these variables, and without this stage many true matches would be lost. Secondly, the other variables in each record that are not used in the record matching process should also be checked at this time, in respect of accuracy, completeness and that the range of values are valid.

Add the record headers for blocking and sorting.

In practice, a header record should be created which contains all the blocking and sorting keys. This step is required where the variables in the file are encrypted, since the records can not be blocked and sorted in a satisfactory manner where the variables in the records are encrypted. The unmatched records can be blocked and re-sorted on a number of different keys and re-matched, the appropriate header records may be created as required and the process of re-blocking and re-sorting repeated until only a few new matches are found.

Add any other derived variables.

It is recommended that additional variables like unique record number or person identifying number should be added to each record. The record number will readily and accurately identify the record for further processing, and the person number would be used for the linkage between the two records and all other records for this person.

Selecting the manner in which the files will be input to the matching software.

There are three ways in which the data records can be matched with the master file.

i) One to one matches - this method is used where a data record is to be matched to a record on the master file, and is the method normally used for person or entity matching. This method of matching seeks to establish unique pairs of records, both referring to the same person or entity.

ii) Many to one matching - this method is normally used for de-duplicating a file that contains many identical copies of the same record.

iii) One to many matching - in this method a data record is matched against a master file and all the matches for the person (or entity) are selected. This method would be used, for example, in culling all records for a person who has many events, for example health care contacts.

Block and sort the files

After the editing and cleaning stages have been completed and the type of matching decided, both the data file and the master files need to be sorted into the same sequence. The blocking and sorting of the files is dependent upon whether the data file is to be matched to the master or it is to be matched to itself.

a) Match a data file against a master file.

There are two methods of doing this depending on whether records on the data file are to matched against themselves or not and these are: the two file method and the one file method. In the two file method the data and master files are held as two separate files, blocked and sorted in the same manner. Every record on the data file is recursively matched against the records in the corresponding block on the master file. The data records will be matched against all the potential records on the master file but will not be matched with each other.

In the one file method the data file and the master file are merged and blocked and sorted into one combined file, and every record in a given block is matched with every other record in the same block in a triangular fashion, i.e. first with the rest,
followed by the second with the rest etc. Every data record will be matched against all the master file records in the corresponding block. At the same pass the data records are matched with each other, and if required, the master file records could also be matched against each other, which would result in data/data, data/master and master/master matches.

b) Match a data file against itself.

In this method the data file and the master file are merged together, blocked and sorted and matched using one file method described above.

4.2 Select the matching method

The two main methods from which to choose are: the Exact Method and the Probabilistic Method (Sections 6.1 and 7.1). The selection of the appropriate matching method depends on a number of factors, the most important of which are the availability, stability and uniqueness of the variables in the dataset. The general guidelines for the choice of the matching method are:

a) If the records on both the data file and the master file have a matching key on every record, which is stable, has low error, with high discrimination, the most appropriate method to use will be the EXACT METHOD.

b) If a single, unique key does not exist, but there are a small group of high quality partial identifiers, which in combination form a unique variable, the EXACT METHOD can still be used.

c) Where the data is noisy and contains random errors, but there is an array of partial identifiers that could be used for blocking and record matching, the most appropriate method is the PROBABILISTIC METHOD.

4.3 Match the file

The file(s) are matched using the methods described above. In the use of exact matching there are only two possible outcomes; either the records match or they do not match. Where the keys are absent or partially present, it is unlikely that the records will match. Where a combination of partial identifiers are used for the exact match or almost an exact match, clerical intervention will need to be undertaken in an attempt to improve the overall match rate, or the query records can just be left unmatched.

Where probabilistic record matching is used, a third category of outcome is generated, that of a query match. This category is best printed out for clerical intervention, however results drawn from this match which can then be used for trimming the match acceptance threshold (see Section 7.10), or for re-running this data set, and using the new thresholds for this and future data sets. (Section 7.11). Any corrections to the computer matching arising from the clerical intervention should be applied at this time.

The matched output will consist of those data records that matched well with the records on the master file, those records that did not match, and those records which are possibly matched and will require clerical scrutiny. The possible records are printed out in a format so that the clerical staff can inspect the results from the computer run. Where the clerks are sure that the two records (one data record and one master file record) belong to the same person, a correction is applied, and where any doubt exists the records are left unmatched.
4.4 Link the matched records to the rest of the file

The matched data records are merged and then collated with the master file, or the results incorporated into the database index. The matched records should be inserted into the proper temporal sequence, any overlaps or other logical inconsistencies should then be checked at this stage. Where there are logical inconsistencies, for example, hospital discharge after death or the person matched as being in two types of care at the same time, both the data record and all the master file records for this person should be extracted for clerical intervention, since the sequence could be due to bad matching from a previous run which could contain records for a different person. These errors usually result from poor data quality or where two different people have almost identical identifying sets, for example in the case of same sex twins. Other problems arise where the persons are elderly and have changed their names several times, or where the identifying set furnished by the patient is different from that supplied by the next of kin on a death record.

4.5 Build the linked file and check for errors

The result of matching one record against another record will be one of the following four outcomes:

a) **Good match.** The records have matched together and refer to the same person or entity. A small random sample of the matches should be checked for accuracy in the matching, especially for those matches that are near the threshold cut-off values (see Section 7.11).

b) **No match.** The records are deemed to belong to different people or entities and do not match together. A small random sample of these non-matches should be checked for accuracy of the matching, especially those matches that are near the threshold cut-off values.

c) **False negative or Type I error.** This type of errors arises when the records which should have been matched together have not been matched because the overall matching criterion falls below the preset threshold and so the records are left unmatched. These records could be matched in the clerical step, or each could be matched to a third record in a subsequent match run, and in this way be matched together (see Section 8.1).

Blocking and sorting the files on a number of different keys and re-matching will identify the record pairs that failed to be linked. A good example is, where a person has changed their surname or forename and not been linked using these keys. The link could be made at some future run using either their date of birth, address, or some other ubiquitous variable. As each match proceeds, the number of unlinked pairs should diminish and an estimate of the number of potential but unlinked pairs can then be made. Another approach is to cull the records for those people who are expected to have a further record, for example a health care record and a transfer to another unit, or where the person has died and a death record should have been linked to the record set.

d) **False positive or Type II error.** This type of error arises when the records for two different people have been matched together, possibly due to having very similar matching variables. Suitable logical methods in the linking step should find these erroneous matches and break the links, otherwise sampling the matched file will have to be used to estimate the false positive rate.

There are two methods of estimating the numbers of badly matched records. The first method, is to randomly sample the linked file and cull the whole set of records that have been linked under the same person number. The set can then be clerically examined to determine whether in fact all the records belong to two or more different
people. This could apply to the records for twins and other multiple births where the identifying variables are almost the same. The second method is to examine all the records for people who have large numbers of events on the file and check for inconsistencies, for example, changes in common items like date of birth, gender or first forename which are usually regarded as stable.

4.6 Other considerations

When setting up a large scale matching project there are a number of other considerations that need to be taken into account. Guidelines on the choice of options and how best to proceed are given below:

a) Matching files using names information

The names may belong to a person, a business or an organisation. It may well be the case that names have changed, contain spelling errors or have been truncated in transcription. The pragmatic approach would be to use an exact matching method, followed by probabilistic matching, followed by the clerical method. For high quality person files, match rates using the exact method can achieve 75–85 per cent while using exceptionally poor quality person files it can be as low as two per cent. The typical match rate for business data files is 20–30 per cent. The remaining data records can be matched using the probabilistic matching method.

b) Matching files containing centrally allocated personal numbers

Although the best method for matching these datasets is the exact match, there is a small chance that the check digit methodology or the number allocation procedures may fail. In this case it would be prudent to use some other variable in addition to the matching key for verification purposes. The variable should be a unique and personal one, and may be selected from date of birth, age, postcode, other national or locally allocated number, or account number.

c) Online or batch processing

The Exact Matching Method and the Probabilistic Matching Method can be implemented off-line (batch mode) or on-line. For example, when matching a data record to the NHSCR register, one may implement these methods in batch mode, using the unique register number in an exact match, or using the name information in a probabilistic match. Alternatively, these methods can be set up in on-line mode. With the operator sitting at a computer terminal, the exact match would be the preferred method, where the selection of a good match is from a list displayed on a terminal. Although the register is indexed using the NHS number, the majority of the enquiries are based on name and address. When the search fails, the next choices would be to use either a search using the Soundex of the search term or a wild-card search.

d) Number and size of files to be matched

The number of records in the files determines the method of blocking. In exact matching the size of the files is irrelevant since the method is based on comparing one key. In probabilistic matching, if the files are very large, it would be both impractical and uneconomic to match the whole data file against the master file. Blocking methods would need to be implemented (see Section 5.3) to control the size of the blocks, consistent with the objective that all potential match pairs end up in the same block. For files containing up to 1 million records a single variable can be used for blocking that will be efficient for matching. For larger files, a combination of two or more variables will be required to create the blocks.
5. **Data editing and parsing for matching and linkage**

The data file and the master file are often compiled on different occasions and in different environments and are often in different formats. Before the files can be matched, it is important that the two files are converted into a common standard format and any errors that can possibly affect the matching process are edited out. Further, the size of the files may be very large and direct comparison of each of the records on the data file with the records on the master file may be prohibitively expensive even with the modern and powerful computing systems. The two files need to be partitioned into manageable blocks. This partitioning of the data files into mutually exclusive blocks is based on a selected set of key matching variables constructed from the individual matching variables.

This section concentrates on methods and techniques employed to pre-process the input files to be ready for matching. These mainly consist of:

- selection and definition of matching variables,
- editing and parsing the matching variables,
- blocking and sorting of the files, and
- the structure and organisation of files.

The details of these operations are described in detail in the following Sections.

### 5.1 Selection and definition of matching variables

#### 5.1.1 Selection of the matching variables

The choice of the matching variables depends on the type and contents of the data file and the master file. The main requirement is that any variable selected as a matching variable must occur both on the data file and the master file. The precise definitions of the matching variables, and the quality of reporting and capture of the these are crucial for the success of the linkage process.

If the data and the master files already exist, potential matching variables must be evaluated to determine whether a linkage of acceptable quality is feasible or not. If one or more of the input files have yet to be created, it may be possible to influence file development in ways that will assist record linkage, e.g. by adding variables needed for linkage to other files, and by using operating definitions and formats that are compatible with those files. Kasprzyk (1983) describes procedures developed to maximise the completeness and accuracy of reporting of social security numbers in the Survey of Income and Program Participation (SIPP) (his purpose was to facilitate enhancement of the survey results through linkages with various kinds of administrative records).

To illustrate the choice of matching variables and the issues involved, let us consider the problem of linking a file of person records based on common identifying information: e.g. the NHS number, names, address, postcode, gender, and birth date.

National Health Service (NHS) numbers are issued at a person’s birth registration or when they have registered as an immigrant with the NHS. The new-style NHS numbers have been issued since 1997, and have been back-loaded into General Practitioner, District Health Authority and hospital trust master indexes and it expected that the NHS number will become a nearly universal identifier. The NHS number (subject to rigorous confidentiality constraints – see Section 12) is one of the most important linking variables that can be used for person
matching purposes, since it is issued to almost all of the population of England and Wales (and issued separately in Scotland and Northern Ireland). Since it has an embedded check-digit it is accurate and stable enough to be used for exact matching. The presence of the internal check digit in the NHS number enables the identification of errors by a simple examination of the number itself. (see Annex D). In Scotland the Community Health Index number is also used in the record linkage process.

All babies born in the UK are issued with the new NHS number as part of the birth registration process. If the baby (and its parents) leave the country for a long period of time (say 6 months) on re-entering the UK they will be required to re-register with the NHS and a new set of NHS numbers will be issued. The NHSCR has generated some duplicates in the allocation of the new NHS number from the old numbers since some people may have two or more numbers possibly as a result of changing their marital status or registering with two different GPs in two different health districts. Because of the low frequency of the problem, multiple NHS numbers can largely be ignored unless one is looking at longitudinal information stretching back to the early days of the NHS.

How can matching be achieved where the new NHS number is either missing or proves unusable? We need to link records using other matching variables. As a general rule, none of the other variables are unique and all of them are subject to errors and omissions either in reporting, transcription or keying. The identifying variables can be considered in the following six quite separate groups:

**Group 1** - consists of the proper names of the person which rarely changes during the lifetime of a person, with the exception of change of present surname where women traditionally adopt their husband’s name on marriage. The readily collected names are birth surname, present surname, first forename or first initial, second forename or second initial, other forenames.

**Group 2** - consists of the non-name personal characteristics that are fixed at birth and very rarely change during the course of a person’s lifetime. It includes gender (sex at birth), date of birth, birth order (in the case of multiple birth), place of birth (address where parents were living when the person was born), NHS number (allocated at birth registration), CHI (community health index) number allocated in Scotland, National Insurance number (NI), ethnicity.

**Group 3** - consists of socio-demographic variables that may change many times during the course of a person’s lifetime: street address, postcode, general practitioner, marital status, social class, unique numbers allocated by a health district, numbers allocated by a hospital or health trust, numbers allocated by general practitioner computing systems, and any other local or national numbers issued by specialised disease or handicap registers.

**Group 4** - consists of variables that may be used in the compilation of special registers: clinical speciality, diagnosis, surgical procedure code, cancer site, drug idiosyncrasy or therapy, occupation, date of death and other dates, e.g. dates of delivery.

**Group 5** - this group consists of variables that may be used for family record linkage: the names described in group 1 together with other surnames, mother’s birth surname, father’s surname, marital status, date of marriage, number of marriages, number of births, birth order, birth weights, dates of delivery.

**Group 6** - consists of arbitrarily allocated numbers that uniquely identify the record (accession number) or this person (matched person number), historical markers for the data in the record, dates, and the edition or version of the codes used within the record.
It is normal practice to match together records that contain a number of the matching variables selected from one or more of the above groups, normally from groups 1, 2, 3 and 6. It is possible to concatenate two or more of the matching variables to construct a compound matching variable, eg using variables like date of birth / gender / postcode, and use it in the exact match. This enables the user to exploit the power, speed and flexibility of the exact match without the benefit of having a unique number.

Still other variables could be used for matching, for example, ethnicity, marital status and telephone number. Ethnicity is an variable that has similar characteristics to gender except not nearly as well reported (unless it is recorded as black or non-black). Telephone numbers while potentially of enormous value, are not normally stored in the UK health and administrative systems since some people may not have a home telephone number.

Problems arise when matching variables are missing, partially given or subject to errors. Some of the problems encountered with the use of matching variables are highlighted in the following Section.

5.1.2 Typical errors that occur in matching variables

Errors in the matching variables may creep in during the capture and processing of these variables. Sources of errors in the matching variables are: variations in spellings, data coding and preparation, use of phonetic name compression methods, use of name synonyms and nicknames, anglicisation of foreign names (usually mid-european), use of initials, truncation and abbreviation of names and addresses, use of compound names, missing words and extra words. During data entry, any extra words being entered in a given field can overflow and fill the adjacent field and in this way create extra and erroneous fields.

The errors occurring in the commonly used matching variables are illustrated as follows:

Present surname

As mentioned previously, name changes due to marriage or divorce are perhaps the main difficulty. For some ethnic groups, there can be many surnames and the order of their use may vary. Concatenations of the birth surname and the marriage or partnership name into a compound (or hyphenated name) are common so that both parts are required for matching purposes. Spelling variations are quite common in surnames due to the effects of transcription of the names through the various systems. In some cultures there is no exact equivalent of a surname, and members of a family may have no single name in common. It is common for Sikhs to add Singh to their name sequences while the addition of Kaur often means that the person is an unmarried girl although in other circumstances it can equally apply to married women.

Other surnames

People may have many different surnames (including their birth surname) and the name they use may vary from occasion to occasion. This practice is becoming more common in the UK, with the increasing number of single parent families. For example, an unmarried mother may give her birth surname when registering birth and her common-use surname, which may be that of her partner, when registering with a doctor. And it is this latter registration which generates the NHS number. Consequently, this creates problems when linking birth registrations to NHSCR records. Often this variable may not be collected.

First forename

There are wide variations in the spelling of forenames due to recording and transcription errors. It is fashionable to modify existing forenames either to modernise them or to copy the
names used by television or sports stars, for example (Rebecca becomes Rebekah). Other widespread problems include the use of nicknames and contractions. Some are readily identifiable (Jim for James, Wm for William, Liz for Elizabeth) but others are not (Ginger for Paul, or Blondie for Jane). Some records may just record the fact that the person is a baby, or a twin, and until such time as the birth is registered, the records may contain the name BABY or TWIN.

Other forenames

People may have many forenames, and the forenames that they use may vary from time to time. Elderly people in particular may have a large number of forenames, some of which are recorded in hospital computing systems while others are recorded at the person’s death (normally supplied by the next-of-kin). Where the person is young and living at home the differences in the forenames can be used to detect twins, i.e. the only variables that differ are the forenames. Where the person is much older, it is more difficult to decide whether the two records are for the same person if the only variables that are different are the forenames. It is difficult to decide whether the person is using a number of different forenames at various times or as their circumstances change. Without further investigation the match between such records must be regarded as dubious and further information sought.

Address

This is an excellent variable for confirming otherwise questionable links. Disagreements are hard to interpret, however, because of address changes, address variations and differences between mailing addresses (usually all that is available in administrative files) and physical addresses (generally all that is obtained in a household survey). Research on this variable has been done by Childers and Hogan (1984).

Postcode

This is another excellent variable for confirming questionable links. Since it is derived from the street address, however, the same comments will apply as for address (above).

Gender (sex)

Gender is generally well reported and, except for transcription and recording errors is a very reliable variable. The main difficulty is that gender may not always be available in some administrative records. For example, some databases do not collect this variable and it can only be generated through the recoding of forenames, which cannot be done with complete accuracy.

Date of birth

The day and month of birth are generally well reported even by proxy respondents. Year of birth can be used with a tolerance to good effect as a matching variable. If the variable is collected along with the present age of the person, checks can be performed at the data collection stage to verify that the year of birth and the age agree. There are problems where the DD and the MM have been transposed from the European format (DMMYY) to the US format (MMDDYY) largely through the use of US computing systems and software. With the increasing use of hospital master index systems this error is now quite small. In some cultures there is no concept of date of birth and in this case the date is recorded as 0101 YEAR in which they were born. Problems also arise where parts of the date of birth are missing or not known: for example, missing day of birth could be substituted by using 01, 15 or 99, or where both day and month are missing the possible default values are 0101, 1506, 3006, 0107, 1507 or 9999.
Other problems encountered in the use of the matching variables are:

a) Swapping of names and surnames

Occasionally the surnames and forenames are swapped around, and there may be spelling variations in the names due to transcription and data preparation errors. If the dataset is matched against a population index it should be possible to improve the quality of the dataset. For example, it might be possible to get extra matching variables like second forename, obtain better spellings for the names, or the order of the names can be reorganised (switch forenames and surnames), or tests made for migration or death.

b) Key variable truncated

Where some of the key matching or blocking variables are missing, partially recorded or not known, they can be replaced with a standard default value, examples of which are shown in Box 5.1. Where a variable has been partially recorded due to some limitation in the transcription process or in the computing storage allocation, methods need to be devised to compare the corresponding data and master file variables which may have strings of different lengths. A good example of this is where the forenames have been captured at different times and stored using different storage standards. The forenames may have been recorded in a 1 to 6 character field, through 8 and up to 12 or even higher.

<table>
<thead>
<tr>
<th>Box 5.1 Examples of variables that are set to unknown values</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dates:</strong> set to 0101YY, 010199, 999999 etc or for date of birth 0101YY, 1506YY, 3006YY, 0107YY, 1507YY, 0101YEAR etc</td>
</tr>
<tr>
<td><strong>Names:</strong> set to spaces, NK, UNKNOWN, or ZZZZ BABY, MALE, FEMALE, TWIN, TRIPLET, INFANT etc</td>
</tr>
<tr>
<td><strong>Other variables:</strong> set to 9, 99, 9999, -1 etc NK (Not Known) NA (Not applicable) NC (Not coded) U (Unknown)</td>
</tr>
</tbody>
</table>

Where forenames have been truncated and stored, say, as six characters, are then compared and matched with the full forename, for example:

CHRISTINA stored as CHRIST (truncated to 6 characters)
CHRISTIAN stored as CHRIST
CHRISTOPHER stored as CHRIST

Only the first 6 characters can be used. The common stem (CHRIST in this example) can only be compared with the same number of characters in the full name.

c) Variables have low discriminating power

Where the records contain a combination of the most common forenames and surnames, for example, the surname is SMITH or JONES and the forenames are JOHN or MARGARET, the name identifiers alone cannot provide a large enough measure of discrimination to separate two JOHN SMITHs. The outcome weights will reflect this and consequently will be allocated a low value. For the match to be accepted, other ubiquitous variables are needed to
bolster the total outcome (binit) weight so that the total weight may exceed the preset threshold value and the match can then be accepted (see Section 7 for more details).

Some of the names recorded in the forename variable can be applied to a wider group of people, and are non specific, thereby having a low discrimination. The examples presented in Box 5.2 show the different kinds of names that have a low quality for the purposes of record matching:

**Box 5.2 Examples of names or titles that have low discriminating power and will therefore generate low outcome weights**

1. Names that contain certain cultural titles, for example: Bibi, Kaur or Singh, etc.

2. Temporary names given to babies or very young children in hospital, for example: Baby, Babyone, Babytwo, Boy, Girl, Twin, Twin1, Twin2, Triplet, etc.

3. Names given to patients in order to preserve their identity, for example: ZZZZ, anon, anonymous, anonymised patient, unknown patient.

When comparing these name strings, they should be identified using the parsing procedures (see Section 5.2) and then allocated a very low outcome weight by the system, otherwise all the babies born on one day could possibly be matched together.

*d) Embedded titles in the names variables*

The surname and forename strings may contain embedded titles, such as those presented in Box 5.3.

**Box 5.3 Examples of titles that are usually embedded in or appended to the surname or forename variables**

- Marital status: Mr, Mrs, Ms, Miss, Master, Son, Daughter etc.
- Peerage: Lord, Lady, Baron, Viscount etc.
- Civil Honours: Hon, Sir, Dame, Lady etc.
- Political titles: MP, Councillor, Mayor etc.
- Academic Title: Dr, Professor etc.
- Academic Degree: MA, MD, PhD etc.
- Military: Major, General, Lt.Col etc.
- Church: Bishop, Monsignor, Rabbi, Father, Brother, Sister etc.
- Family order: Fred Jr, Bill Sr, Hiram 3, Hiram Third or Hiram III etc.
- Hyphenated names: Baden-Powell, Twistleton-Fiennes etc.
- Concatenated names: Joan Brown Smith etc.

Source: ORLS, 1999

Before the name string can be used for file blocking and record matching, the string needs to parsed (see Section 5.2) and the various components identified and separated. Some of the titles can be used for linkage purposes, for example those associated with marital status, academic title and hyphenated names.
e) **Missing or partially given matching variables**

In some cases the matching variable may be missing or set to some arbitrary value. It can still be used for matching provided that the outcome weight and the match threshold is adjusted accordingly, for details see Section 7.11. It sometimes happens that one or many of the matching variables may be partially complete. For example, only the first six characters of the forename or just the initial may have been recorded. The name variables may have been set to: spaces (or blanks), or some agreed default value for not known or not collected. The corresponding matching variables in both the records can then be compared and the amount of agreement computed. Where there are small deviations in, say, the date of birth the amount of agreement can still be computed, based on empirical results compiled from previously clerically checked matches using sample or similar datasets.

### 5.2 Editing, parsing and standardisation of the matching variables

The matching variables are used to define matching keys which partition the input files into blocks. Due to the problems associated with the use of matching variables, these may not be suitable as such. Therefore rigorous editing and parsing of the matching variables is undertaken in order to minimise errors. Without it, there is the possibility that many true matches would erroneously be designated as non-links since the identifying information could not be adequately compared.

Another advantage of editing and parsing is standardisation on the definition and representation of the matching variables. Practitioners undertaking matching and record linkage must pay particular attention to the formatting and standardisation of the matching variables since it can improve the chances of correctly linking two records. In some cases, record linkage would be impossible without the initial re-formatting of records. Consider matching two files using names where one of the files has surnames first and the other has forenames first. Differences in the presence or absence of titles (Mr. Mrs. Ms. Dr. etc.) can also cause difficulties, as well as the use of Junior/Senior (Jr/Sr) etc (see Section 5.2). The standardisation can result in some kinds of distortion (truncation of names down to a common and standard length) and loss of information which will increase the likelihood of designating some pairs of records as positive links when, in fact, they do not match.

The main elements of the editing and parsing process are, (i) parsing and standardisation of the matching variables, and (ii) creation of linkage files.

### 5.2.1 Parsing and standardisation of the matching variables

DeGuire (1988) presents the concepts needed for parsing and standardising addresses but the same techniques could also be applied to the parsing of names. The procedure for parsing and standardisation of the matching variables involves identifying the constituent parts of the matching variables and representing them in a common standard way through the use of look up tables, lexicons, and phonetic coding systems. The standardised individual elements are then rearranged into a common order. The parsing and standardisation of the commonly used matching variables are detailed below.

a) **Standardisation of surnames and forenames**

The basic uses of standardisation are; firstly, to replace the many spelling variations of commonly occurring names and addresses with standard spellings such as fixed abbreviations or correct spellings, and secondly, to use the key words generated during the standardisation process as hints for the development of editing and parsing subroutines. The purpose of name standardisation in record matching is to allow name matching software to work more efficiently by presenting names in a consistent fashion and by separating out parts of the name that would have little or no value in matching.
In the standardisation process, forename spelling variations such as LIZ and BETTY might for consistency be replaced with the original or formal spelling such as ELIZABETH. It is also possible to convert identifying stem words such as FRED although these could equally be associated with ALFRED and FREDERICK. Other procedures sometimes used in formatting names include the removal of punctuation or blanks. For example: O'BRIEN becomes OBRIEN, Le MESURIER becomes LEMESURIER and Van DAMM becomes VANDAMM. As previously described, dictionaries and lexicons have been developed that can relate commonly used nicknames and name contractions to formal names (BOB and ROBERT, BETSY and ELIZABETH) and to link the common variations in spelling (SMITH, SMYTH, SMYTHER or HORTON, HAWTON, HOUGHTON).

Sometimes, the order of surnames and forenames in a record is not clear, there may be records in which the names and surnames have been swapped, for example JOHN SMITH recorded as SMITH JOHN. The ORLS have developed lexicons for identifying most common forenames and surnames and use it for storing or checking the name against the type of field. Problems occur where both the forename and the surname can be used equally as a surname or forename, for example JAMES DOUGLAS could be DOUGLAS JAMES or JIM LORDE could be LORD JIM. During the data capture and preparation phases, the order of some names may not recorded in a consistent fashion. The ORLS matching algorithms will match the names in the two records against each other, and the best fit (usually measured using the highest outcome weight) will be used for the matching process.

Use of look-up/conversion tables, lexicons and dictionaries

While a conventional dictionary primarily provides definitions of words and phrases, it may also be used to provide a list of synonyms. This property may be exploited in the conversion of a given name to another name, i.e. a synonym for the given name.

The use of a dictionary or lexicon to retrieve names or words similar to a given word has many applications in spelling-correction systems, and used, for example, in word processors to correct misspelled or miss-keyed words in a document, or to retrieve a forename from a list of nicknames or contractions (see Box 5.4). This technique provides a good method for comparing names in both exact and probabilistic matching techniques.

Box 5.4 A sample of a Lexicon for converting names to a formal name

| WILLIAM   | WILLY, BILLY, WILL, BILL, WM etc. |
| JOHN      | JOHNNY, JON etc. |
| ELIZABETH | LIZ, LIZA, BETTY, LIZZY, BETH, BETSY, ELISABETH etc., |
| MARGARET  | MAGGIE, MAGGY, MEGGIE, MEG, PEGGY, MADGE, MARGE etc |

Source: ORLS, 1999

The process involves two stages. The first stage involves checking the spelling of the name against all the relevant entries in the lexicon and if it is found to be valid, the second stage returns the various synonyms for use in phonetic coding schemes or table look-up methods. Errors that have occurred when the original valid name was transformed using the lexicon into a different valid name (for example ELIZABETH from LIZA), are inherently undetectable in this type of scheme, and therefore it is non-reversible.

Lexicons can also be used for the conversion of a name string to a code, or a service name to a service code. A sample of such a lexicon is presented in Box 5.4.
An example of differences in name spelling that may be used for the construction of a lexicon used for equivalencing the various spellings of common forenames is shown in Box 5.5.

A common procedure for parsing surnames has been to encode surnames phonetically and to use the encoded values, usually along with other matching variables, for blocking the file. Although the phonetic code of the names are used for positioning both the data file and the master file, the original names on both the records are used for the matching process.

<table>
<thead>
<tr>
<th>Box 5.5 Some typical variations on the spelling of ABIGAIL</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABAGAYLE (1)</td>
</tr>
<tr>
<td>ABBEYGAILE (1)</td>
</tr>
<tr>
<td>ABBIEGAE (1)</td>
</tr>
<tr>
<td>ABBYGAIL (3)</td>
</tr>
<tr>
<td>ABBYGEL (1)</td>
</tr>
<tr>
<td>ABEGALE (1)</td>
</tr>
<tr>
<td>ABIGAEL (2)</td>
</tr>
</tbody>
</table>

The number in brackets indicates the frequency of the name on the ORLS master files.

Source: ORLS 1999

Use of phonetic coding schemes

The two procedures in common use for the phonetic coding of names are the Russell Soundex Code and the New York State Identification and Intelligence System (NYSIIS). The Soundex system, which is the older of the two, creates a four-character alpha-numeric code which uses the first letter of the surname for the first character of the code and a further three digits (Lynch and Arends, 1977). (see Annex B)

The NYSIIS code is a fixed length alphanumerica code that divides the population of North American and European surnames into groups which are smaller in size than those associated with the Soundex codes (Lynch and Arends, 1977). Both coding systems are designed so that surnames of similar sound have the same code and frequently encountered errors of reporting do not cause changes in the code (Howe and Lindsay, 1981). The primary objective of the use of phonetic coding is to create blocks which:

a) places all variations in spelling of a given surname into the same block
b) limits the size of the codes, that is, limiting the number of records assigned a given code

creates codes that contain few dissimilar surnames
d) requires minimal computer processing without the need to employ large look-up or conversion tables.

Most of these systems have two features in common:

1) The vowel information is either partially or wholly suppressed because of its instability. Some phonetic algorithms retain the position of the vowel in the names string but convert them into the letter 'A'.

2) Certain consonants with similar sounds (or groups of letters containing these consonants) are replaced by a standard character or group of characters representing that sound or phoneme. For example the letters M and N are grouped together as are D and T.

The Soundex and NYSIIS algorithms (see Annex B) possess the above features. However, NYSIIS retains information on the sequence of and position of the vowels in the name by changing them all to the letter 'A', whereas Soundex removes of the vowels. All of the
algorithms are capable of revealing similarities between names even where the coded forms do not agree precisely. Finally, where shortened forms of the names are needed for compactness, name compression may be desirable because it loses only the vowels and the redundant consonants.

A development undertaken by the Oxford Record Linkage Study (Gill, 1986, 1993, 1997), referred to as the Oxford Name Compression Algorithm (ONCA), uses an anglicised version of the NYSIIS method of compression as the initial or pre-processing stage, and the transformed and partially compressed name is then Soundexed in the usual way. This two-stage technique has been used successfully for blocking the files of the ORLS, and overcomes most of the unsatisfactory features of pure Soundexing while retaining a convenient four-character fixed-length format.

The file blocks produced using the ONCA procedure, in common with the other phonetic compression algorithms vary in size, from quite small and manageable for the less common surnames to very large and uneconomic for the more common surnames. Further sub-division of the phonetic blocks on the file is usually effected using: gender, first initial or date of birth either singly or in combination.

During the course of a number of studies on mortality, it was found that the number of links lost through problems in the method of blocking the files using names, ranged from two per cent for most studies through to 25 per cent where the identifiers were poor (Lalonde, 1992). For this reason, the files may be blocked by the phonetic code, and the matching is then carried out only between records that fall within the same coded blocks. This method offers a very efficient method of matching small blocks that contain most of the possible variations of the surname.

Occasionally, phonetic coding is used to reveal similarities between names even where the codes themselves may differ. A third use of such codes to reduce the physical sizes of the names is rarely undertaken, but it is used extensively for the preparation of constant length blocking keys.

b) Standardisation of business names

The main difficulty with business names is that even when they are properly parsed, the identifying information may be indeterminate. Sometimes the pairs of names can refer to the same business although the names can be quite different. On other occasions the names can be quite similar but the businesses are very different. Because the names information may be insufficient for accurately determining the status of the match, address information and other identifying characteristics may have to be obtained using clerical procedures. If the additional address information is indeterminate, then at least one establishment in each pair may have to be contacted.

c) Standardisation of addresses

Standardisation of addresses operates in a similar fashion to standardisation of names. Abbreviations like 'Rd' or 'Cres' should be replaced by appropriate expansions to 'Road' or 'Crescent' or to a set of standard abbreviations commonly used by the organisation. For example, when a variation of a rural address (e.g. Hill Top Farm or Sunny-Side Nursing Home) is encountered, the software should use a set of parsing routines different from those associated with house-number/street-name addresses. Where reference files containing town and county, and postal codes are available from the post office or from some other source, the town names in the address lists can be reformatted into a standard form that is consistent with the reference list.
Parsing divides the free-form address variable into a common set of components that can be compared, for example, street number, street name, town and county. Parsing algorithms often use words that have been standardised. For example, words such as 'STREET' or 'ROAD' would cause parsing algorithms to apply different procedures than words such as 'HIGH' or 'OXFORD.' While exact, character-by-character comparison of the standardised but unparsed names would yield no matches, use of the components in the address might help designate some pairs as links. Commercial software packages such as PAAS software (DeGuire 1988) are excellent at parsing and standardising of addresses. The unresolved cases can be parsed manually since human beings are best at comparing the many types of addresses because they can associate the corresponding components in free-form addresses.

d) Standardisation of postcodes

The UK postcode is issued and supported by the Royal Mail and was originally designed to route the delivery of the mail. In recent years it is also used as a proxy measure of geographic location in the absence of other measures, given that there is a National Grid Reference System used in the generation of the Ordnance Survey maps, although most people are unlikely to know their grid references. The postcode consists of two parts, the inward code which designates the postal town and the outward code which designates the postman’s round.

The format of the postcode consists of one or two letters which designate the main postal town, followed by two or three numbers then two further letters. The code can be between five and seven characters long, and sometimes a space character is inserted between the inward code and the outward code, for example OX3V7LF (where V denotes a space character). Errors occur where the code, for example OX12V9EL is truncated to fit into a seven character variable, instead of removing the embedded space character, the last character is normally chopped off, resulting in OX12V9E. It is recommended that all postcodes used for matching purposes should have the embedded space character removed and the whole code left justified and stored in a seven character wide variable.

\[
\begin{align*}
\text{OX12V9EL} & \rightarrow \text{OX129EL} \\
\text{B1 2SP} & \rightarrow \text{B12SP}VV \\
\text{(Where V denotes a space character)}
\end{align*}
\]

5.2.2 Creation of linkage files

Creating additional records on the master file where a number of different names are recorded in a person’s record

In systems that use real names, the process must be capable of matching together all the different surnames or forenames that any person may supply over the period of the file. In most cases the need to change a name will arise from marriage, divorce or deed-poll changes; or in the case of businesses, change of the registration details. Removing the old name from the system cannot be done, since such records will contain historical data. To maximise the ability to match records, the most effective method is to add the additional names entries to the index or master file for the same person or entity, and it might also be advantageous to indicate which record contains the current preferred name or registration. This is analogous to adding extra cards in a card index, one under the present surname and the other under the birth surname, and so on.

The fundamental decision in creating the blocks for record matching, is whether the blocks based on the phonetic code of the names contain all the possible versions of the names for a given person. Where the phonetic codes are different, the records may be spread over many
different blocks. In this case, organisation of the file is more difficult, and the only solution is to create multiple copies of records (one in each of the phonetic blocks) or multiple pointers to each record.

The multiple records on the master file are generated from all the combinations of present and birth surnames, and forenames. To illustrate the generation of such extra records consider the example presented in Box 5.6.

Mrs Hall would have a master file record included in each of the eight ONCA/Initial blocks. A data record containing any combination of the above names would generate an ONCA/initial code similar to any one of the eight above, which would then have a high probability of matching to any of the name variations at the matching stage. All eight entries on the master file, as shown in the example presented in Box 5.6, would have the same person number and accession number, since they are exact copies of the original record.

*Record headers and other variables that may be added to each record prior to matching*

It is recommended that a header is added to every record on the data sets. In the ORLS system; this improves the efficiency of file handling, blocking and sorting, and the file header contains those keys that will be used for the various matching and decryption processes. These are described in Box 5.7.

*Use of a record header when record matching using an encrypted file*

The ORLS have developed a record matching system that has the flexibility to match records in which the names and other identifiers have been encrypted. This is done in two stages, in the first stage a header record is generated for every record on the dataset, in which is stored all the blocking information as shown in Box 5.7. The data in the record header is computed from the encrypted names and date of birth information, and the resulting codes are generated and stored as plain text while the variables in the rest of the record are left encrypted. Using this header, the records can be sorted in the appropriate blocks in both the data file and the master file. As each record is read from the input files into the computing system it is decrypted and stored in the internal buffers of the matching program. Only in this way are the decrypted text identifiers made available to the record matching program. At the end of the matching run all the buffers are cleared. The problem then arises about the security and storage of the computer listings which are generated for clerical checking. These listings should be stored in a locked environment and shredded and pulped when all the clerical checks have been completed.
Box 5.6 Creating extra master file records where there are many variations of the surnames and the forenames

Consider the record for a woman who has the following names information:

Birth surname: SMITH
Present surname (married surname): HALL
First forename: LIZ (contraction of Elizabeth)
Second forename: PEGGY (contraction of Margaret)
Year of birth: 1948 (old enough to be married)

Where the file blocking is based on the phonetic code further divided by the first initial, eight identical records would be generated on the master file and each record indexed under the various combinations of the phonetic code (in the ORLS case the ONCA code) and first initial, as follows:

<table>
<thead>
<tr>
<th>Blocked under the Present Surname</th>
<th>HALL: i.e. (ONCA H400), for Liz</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>H400L for Elizabeth (formal version of Liz)</td>
</tr>
<tr>
<td></td>
<td>H400E for Peggy</td>
</tr>
<tr>
<td></td>
<td>H400P for Margaret (formal version of Peggy)</td>
</tr>
<tr>
<td>Blocked under the Birth Surname</td>
<td>SMITH, (i.e. ONCA S530), for Liz</td>
</tr>
<tr>
<td></td>
<td>S530L for Elizabeth</td>
</tr>
<tr>
<td></td>
<td>S530E for Peggy</td>
</tr>
<tr>
<td></td>
<td>S530P for Margaret</td>
</tr>
</tbody>
</table>

Addition of historical pointers/markers

In practice, the ORLS have found it advantageous to add three further variables to each record: firstly to allocate arbitrary numbers to each record, and secondly, to identify the coding systems used in each record:

**Accession number.** An arbitrary number allocated from a pool of such numbers, and is absolutely unique to this record. The number is never ever changed and is used for the identification of the record during the correction and amendment stages. It is also used as the main key where the files are separated into a names and address file (which contains no administrative or clinical data), and an analysis file (which contains no names or address data). In the ORLS system this number is check digitized using the modulus 97 (see Annex D).

**Person or system number.** A arbitrary number allocated from a pool of such numbers. This number should be the same on all the records which belong to the same person. The number can be changed or replaced according to preset rules where the data file record matches with the master file record (see Section 8). This number is check digitized using the modulus 97 (see Annex D).

**Coding editions.** Indicators that record the various editions of the coding frames used in each record, for example the version of the ICD (International Classification of Diseases) or the surgical procedure codes. These indicators ensure that the correct coding edition is always recorded on each and every record and reliance is not placed on vague ranges of dates, or other markers.
Box 5.7 Details of the ORLS record headers

1. The primary name blocking keys are generated using both the ONCA and the NYSIIS compressions of the present and the birth surname.

2. The other primary keys are based on the full date of birth (CCYYMMDD), gender, the first forename or the postcode.

3. Where the birth surname is present on the record and is not the same as the present surname, as would be the case for a married woman, a further record is generated on the master file under the phonetic code of birth surname and again sub-divided by the initial - (a process termed expanding the file, see Section 5.2)

4. The secondary keys are generated using the gender or the initial letter of the first forename. Where this forename is a nickname or a known contraction of the 'formal' forename, the initial of the 'formal' forename is used. For example, if the recorded forename was Bill, the 'formal' forename would be William, and the initial used for blocking would be W. In practice two records would be generated, the first blocked on the initial B (for Bill) and the second blocked on the initial W (for William).

5.3 Blocking and sorting of the data files

In matching files of any reasonable size it is not possible to compare all the record pairs since the number of possible match pairs is the product of the number of records on each of the files. Even two small files of say n records each would generate n*n pairs to compare. In practice, to reduce the time spent in attempting unproductive matches, each of the input files should be sorted into blocks (sometimes referred to as 'pockets') prior to the matching stage. The records in a block on the data file will only be compared with the records in the corresponding block on the master file. Special provisions may be needed if multiple matches, each with a different blocking structure, are to be performed (refer also to file expansion in Section 5.2).

Consider the matching of a file against itself for the purposes of de-duplication. If the file contained just 1,000 records the number of comparisons would be (1000*1000) / 2 = 500,000 (since the match between record x and record y is the same as the match between record y and record x, therefore only half of the matches are necessary). In this example it is expected that 1000 match pairs would be found and the remaining 499,000 would be unmatched pairs. Where the files are substantially larger, say a modest 1 million records, the number of comparisons would be 500,000,000,000 and so on. This number of comparisons, even on the fastest machine would prove to be uneconomic in terms of manpower and machine utilisation. For example, if the computing system could achieve 1 million matches per minute, the elapsed time to cross match the file of 1 million records would be 347 days.

If the file consisted of 1000 records and these were divided into 10 blocks of 100 records each, the number of comparisons would be 10 * (100 * 100) / 2 = 50,000, or roughly one tenth of the effort required in the example in the previous paragraph. Extending this methodology to the file with 1,000,000 records the corresponding record pairs that require matching would be 10,000 * (100 * 100) / 2 = 50,000,000 or roughly one ten thousandth of
the effort, and using a computing system that could achieve 1 million matches per minute, the task would be completed in 50 minutes.

In practice, a file of 1 million records blocked by date of birth will have about 10,000 blocks of 100 records each and the processing time will be roughly the same as that described in the previous paragraph. The surname (or phonetic compression of it) is not as well distributed as that of the date of birth file and it is expected that there will be some very large blocks and some very small blocks. Using the name distribution in Annex E, and the experience of name matching by ORLS; with a file size of 1 million records, it is expected that:

- 100 blocks with 2,000 records = 200,000 records = 100,000,000 = 100 minutes
- 200 blocks with 1,000 records = 200,000 records = 100,000,000 = 100 minutes
- 1,000 blocks with 200 records = 200,000 records = 20,000,000 = 20 minutes
- 8,000 blocks with 50 records = 400,000 records = 10,000,000 = 10 minutes

Total matches = 230,000,000 = 230 minutes

The elapsed time to complete all the name matching would be about 4 hours.

In this way, the use of blocking methods support and expedite the matching process by reducing the number of potential match pairs that have to be processed in any one phonetic or other type of block. The number of false non-matches will be increased since some of the potential match pairs may have different blocking criteria and would fall outside the range of blocks being compared. To reduce this failure to match, additional match pairs are generated from the given set of records and inserted into the appropriate parts of the file (Section 5.2). Blocking precludes some of the possible matches since only those within the block are matched, the selection of the blocking variable(s) is a very important step in the record linkage process. It is also necessary to check that the blocking variable does not have any unusually large values so creating very large blocks, since these would greatly increase the processing required.

The choice of an optimum set of matching variables is basically a balance between reliability and discriminating power, and these are essentially independent characteristics of any identifier. Reliability serves to keep the non-matching losses down, and discriminating power is needed to keep the amount of computer processing costs down. The definition of best matching variables implies a balancing of one against the other. Usually, some limit will have to be placed on the acceptable number of comparison pairs of records that must be examined by the computer to do a record matching job. Unless both files are small, this precludes the matching of every data record with every record being searched. At the other extreme, a requirement for precise agreement on a set of identifiers would often result in an unacceptably high number of errors in the form of missed linkages. The optimum lies somewhere in between.

Using a larger number of matching variables generally improves the efficiency of the matching stage; however, there is a point at which the errors and omissions in the larger matching set outweigh the benefits of using more variables. For blocking and matching it is preferable to use names variables which have more distinguishing power than using non-names or address variables, and in this way smaller and more efficient blocks are generated.

Since the blocking variable could contain errors and omissions, sometimes it is necessary to re-block the file using other blocking variables in turn. The files can then be sorted on the new keys and re-matched. The matched outputs can be combined with those from matching using other blocking orders and in this way the record pairs may be matched together (see Section 8.1).
5.4 The structure and organisation of files to be record matched

The two file structures that are normally used in record matching are a clustered or blocked flat file and an indexed database. For the latter, record matching will be undertaken between pairs of records in the database and the index then modified to reflect that the records belong to the same person or entity.

The records in a clustered flat file are so organised that those records having similar content defined by some specified criteria are located close together and are readily accessible as a group. Because retrieval from a bibliographic file is inexact and relies on similarity judgements, this clustered structure is widely used in information retrieval work, and is the type of file structure most commonly encountered in record matching and linkage systems.

The development of a clustered file depends on the selection of a clustering or file blocking technique. Using statistical techniques it is possible to assess, on the basis of record characteristics, the probability that a group of records which relate sufficiently to the same person or entity will be clustered together. A fundamental question in forming the clustered file is whether the clusters or blocks have common members or are disjoint. In the latter case, organisation of the file is more difficult and may require multiple copies of records to be stored in the file, or multiple pointers to each record. This is the approach adopted by ORLS for the preparation of an expanded file to cope with records that contain many name variations (see Section 7.3). Maintenance of a file with overlapping clusters or blocks is difficult because of the need to coordinate the multiple copies or reference pointers each time the file is changed or updated.
6. Exact or deterministic methods of record matching

In this Section the focus will be on the methods of matching two records together using exact matching. This is the method of choice when there exists a unique or a near unique identifier (single matching variable or a combination of matching variables) and the quality of data of is relatively high. The method relies on the comparison of the identifier on the query file being matched against the identifier on the master file. In its simplest form, the method has been implemented in the popular software packages SPSS® and SAS® under the option ‘file merging and sorting’.

6.1 Exact methods of record matching

6.1.1 Defining a unique identifier

The main requirement for exact matching is the availability of a characteristic of a person or object that is unique, universally available, fixed, easily recorded and at the same time readily accessible and verifiable (Section 3.1). The perfect identifier would be an integral trait of the person, or numbers allocated to the person by means of a highly reliable matching procedure. A unique identifier could be defined by using a series of numbers large enough to encompass all the members of the population. Many numbering systems have been devised and these fall into three broad groups.

i) Serial numbering systems - in which a unique number is assigned to each individual from a central allocating point. Serial numbering systems have three main advantages: simple to use, easy to automate, and do not depend upon non-unique characteristics of the individual. The new NHS number can be used for this purpose, since it is unique, almost universal and checkable and it is issued from one central allocation point.

ii) Derived numbering systems - the number is derived using the readily available unique characteristics of the person. The advantage of such numbers is that they can be derived at any place and at any time without reference to a central allocation point. The disadvantage is that they depend on the person’s stated characteristics, and there is a risk that two or more people will share the same characteristics.

iii) Composite numbering systems - combinations of the serial and derived numbering systems that use a central allocation point to obtain parts of the number, and the non-unique characteristics to derive the other part(s).

All three types of number allocation schemes are prone to errors in the recording of the numbers, whether by speech, handwriting or keying. To improve the match rate and reduce the errors and missed matches it is desirable to incorporate a checking device in the serial key. One such method is to incorporate check-digits or check-characters into this key matching variable (Wild, 1968; Hamming, 1986; Gallian, 1989; Gill and Baldwin, 1982; Sethi, 1978; Dass, 1984; Brown, 1973; Holmes, 1975) (Annex D).

Where records do contain these unique checkable numbers, automatic matching can be rapid, reliable and inexpensive. Since the file is sorted on the blocking key and only records with identical identifying sets or keys are matched, this results in very fast matching since no near matches are considered.

Where unique numbers or ciphers are not available or not collected by the system, obvious candidates for use as matching variables are combinations of names, date of birth, gender and perhaps other variables such as address or postcode. The exact match will then depend on their use in combination. Much thought has to be given to the order of such partial identifiers,
since the most reliable identifiers should be used for the more significant end of the combination key and the less reliable identifiers used for the least significant end. In concatenating date of birth, sex and postcode, the most reliable and discriminatory identifier would be the date of birth, and the least reliable identifier would be postcode. The order of the concatenated identifier would then be: date of birth/gender/postcode, and the system could then cope with partially complete postcodes.

6.1.2 Blocking, sorting and matching

Both the data and the master files should be ordered sequentially on the values of the single or compound identifier before attempting matching. The reliability and efficiency of the matching procedure is highly dependent on the manner in which the blocking is carried out. The simplest way to compare the identifier on the two files is to use brute-force (BF) algorithm. It consists merely of trying all possible character positions in the text string. For each such position, it verifies whether the characters match at that position. When all the characters and the numbers agree, the two records can be regarded as matched together, and conversely when they do not agree the records belong to different people/entities.

When using unique identifiers, the outcome of the matching process is clear cut, either the records match or they do not, depending upon whether the identifiers match or not. If there are any differences between the two strings then the match is deemed to fail and the two records are judged to belong to two different people. Where the key has been built up from a number of partial identifiers, some relaxation in the matching criteria could be used to bring together pairs that do not match exactly. The criteria for accepting the match would be in the form of a simple Boolean rule, for example date of birth same, sex same and just the first part of the post code (in area code) the same. This relaxation would depend upon the number of master file records with which the data record matches, and could only be resolved using clerical assistance. It is important not to confuse this with the threshold setting used in Probabilistic matching.

6.1.3 Checking the validity of the match

Having matched the file on the unique identifier, the comparison of sex, date of birth and forename will usually suffice to check that the records have been correctly matched and do belong to the same person. Problems will arise, however, in the case of similarly named and same sex twins, or for elderly people who may have a number of surnames (through many marriages) or they may have two or more forenames, synonyms or nicknames. Where the identifier is composed of partial variables like date of birth, sex and postcode, further checking will need to be undertaken using other variables in each record that are almost universally available.

6.1.4 Resolving uncertainties

The match can only determine whether the data record matches with the master file record, or not. When this record is subsequently linked into the master file and fails the logical checks associated with the linking software (see Section 8), the output will need clerical intervention. Where there is any doubt about the match, the data record should be treated as unlinked just as if it belonged to two different persons or entities.

The errors in the data will cause matching to fail and it will show up in the form of non-matches or generate ties. In these cases, it would be more appropriate to use probabilistic matching to resolve these cases, following the use of exact method. It is a common practice to combine the two record matching approaches: exact techniques may be used as a first stage,
producing files that can be analyzed to generate probabilistic weights for further matching. Roos and Wajda (1991) and Wajda et al (1991) provided some specific examples of exact matching (also see Kendrick and Clarke, 1993).

6.2 Risks associated with wrong matching

The major risks in using data that is badly matched is that the data record will be linked up with records for a different person. Since the only matching options available in the exact match are TRUE or FALSE, the creation of a wrong match indicates that either the blocking or matching keys are wrong. Where the matching key is an allocated number, there may be two reasons for this error,

1) either the wrong digit has been recorded in the identifying number as a result of a transcription error and hopefully this should be trapped using the check digit algorithm, or

2) the person has been issued with a number that had been issued previously to another different person.

Where the matching key is built up from a number of partial identifiers, errors or omissions in any part of key will result in collisions and wrong matching. In many cases this bad match can be detected when a logical check is performed across all the records for any given person. Sometimes it is almost impossible to determine whether the record belongs to the person or not. In these cases, the link cannot be made and the data record is then regarded as belonging to two different persons or entities.

A number of major matching applications are concerned with improving coverage in surveys and censuses. In these cases a false negative is a catastrophic error because each nonmatch is added to a list as a new entity.

6.3 Typical match rates

Using keys based on serial numbers issued from some central allocation point and incorporating a check digit, it is estimated that the match rate should be about 95.6 per cent for modulus 11 and 99.95+ per cent for modulus 97 or the use of a double 11 modulus check digit. Some typical results are shown in Box 6.1.

During the course of a number of studies on mortality, it was found that the number of links lost through problems in the method of blocking the files using names identifiers, ranged from two per cent for most studies through to 25 per cent where the identifiers were poor (Lalonde, 1992). Less likely are the false positive matches arising from the exact correspondence between an incorrect number and a correct number, or between two incorrect numbers.
# Box 6.1 Typical results obtained from exact matching datasets containing names or other matching variables

1. Matching names against a telephone directory  
   Exact name matching (Dolby 1970)  
   85% match rate  
   (5 and 80 rule)

2. Primary matching of ORLS dataset  
   Exact name matching (Gill 1987)  
   87-90%

3. Post enumeration survey  
   Exact name matching (Winkler and Thibaudeau, 1991)  
   75-80%

4. De-duplication of NHSCR  
   Exact name on Surname/forename/sex/date of birth (Gill, 1994)  
   75–85%

5. De-duplication of NHSCR  
   Exact matching using the NYSSIIS code (Gill, 1997)  
   90–93%

6. Exact match of a dataset against a part of the NHSCR  
   Exact matching using the NYSSIIS code (Gill, 1998)  
   87–90%

7. Exact matching of an ORLS (HES) dataset against itself using  
   Date of birth/sex/postcode (Gill, Goldacre,McGuiness, 2000)  
   96%

Source: Various
7. **Probabilistic methods of record matching**

The broad principles underlying probabilistic record matching have already been presented in Section 3.2. This Section provides more details about how it is carried out in practice.

7.1 **When is probabilistic record matching used?**

Probabilistic record matching and linkage is a process of collation and assessment of information from two files which consist of records that might belong to the same person or entity. This is of particular value when many of the records do not have universally available and unique identifiers. There are two major complications that may arise:

a) All computerised records are prone to error. These errors may occur because incorrect information has been obtained from the person, or the data has been transcribed or keyed incorrectly. Because of such errors, the records for the same person may not agree, and conversely the two records that do agree may belong to two different people.

b) Parts of the record might be missing, or be encoded in a different way. The missing data can occur in a random fashion, as happens when some data items are unreadable or lost, or the loss could be systematic, e.g. the lack of a second forename for a person who has only one name, or only one name has been recorded by the system.

| Box 7.1 Cases where exact and probabilistic matching are appropriate

| Example 1  |
|-----------------|-----------------|-----------------|
| HALL STEPHEN    | JOHN            | Male 220738 14 High Street |
| HALL STEPHEN    | JOHN            | Male 220738 14 High Street |

| Example 2  |
|-----------------|-----------------|-----------------|
| HALL STEPHEN    | JOHN            | Male 220738 14 High Street |
| HALL STEVEN     | Male 220838 14 High Street |
| p1 p2 p3 p4 p5 p6 |

The examples in Box 7.1 illustrate the types of record pair in which exact and probabilistic matching are appropriate. In Example 1, since each of the six identifiers are exactly the same, the use of the exact matching methodology will report that the two records belong to the same person. If there are no errors or at the most very few errors in coding or transcription, then exact matching can be used. In Example 2, although the records appear to be for the same person, there are a number of differences: spelling of first forename, missing second forename, and different month of birth. Three out of the six variables are different. This pair would generally fail to be linked using exact matching methodology, and a probabilistic method would probably be more appropriate. The general idea is encapsulated in Smith (1984)'s description:

*Agreements of various identifying variables will generally argue in favour of a linkage, whereas disagreements will argue that the records relate to different people. Numerical weights can be used to quantify the fact that rare names, rare birthplaces, and such, carry more discriminating power when they agree than do their common counterparts.*

Source: Smith, (1984)
7.2 Basic concepts of probabilistic matching

To illustrate the basic theory of probabilistic linkage, it is best to start with a simple example. For this exercise, we only recognise full agreement or disagreement outcomes of specific variables within the identifying sets of variables, and disregard all partial agreements. As a further simplification we assume that the identifiers selected for the comparison are always present on both the data file (D) and the master file (M).

Probabilistic linkage uses weights based on frequency ratios, which give an estimate of the likelihood that the records in the pair under consideration are truly linked relative to the likelihood that the record pair are unlinked. Each frequency ratio in a sense represents the evidence in favour of a true match.

\[
\text{FREQUENCY RATIO} = \frac{\text{relative frequency of agreement (x,y) among \textbf{linked} pairs}}{\text{relative frequency of agreement (x,y) among \textbf{unlinked} pairs}}
\]

(for agreement) \[\frac{\text{relative frequency of disagreement (x,y) among \textbf{linked} pairs}}{\text{relative frequency of disagreement (x,y) among \textbf{unlinked} pairs}}\]

where \(x\) indicates the value of the identifier on the query or data record, and \(y\) indicates the value of the identifier on the candidate or master file record.

These are termed \textit{global frequency ratios} (as opposed to output-specific frequency ratios, which will be discussed below) since the identifiers can take any value. These frequency ratios give some indication of the value of the particular identifier for providing evidence of a true link between the records. Boxes 7.2 and 7.4 give some illustrative examples. From Box 7.2, we can see that agreement on surnames is more persuasive evidence of a true link than agreement on forename or year of birth (which themselves are of approximately equal value, having frequency ratios of 88 and 70 respectively). Disagreement on any of these variables is suggestive of a non-link, but the evidence against a link is much less strong. This is probably due to errors in coding these variables. Box 7.4 illustrates, as you would expect, that agreement on sex alone provides very weak evidence for a link, whereas disagreement on sex is powerful evidence for a non-link. These examples illustrate the usefulness of frequency ratios as indicators of the value of agreements and disagreements on particular variables as evidence of a true link and non-links respectively.

To calculate these frequency ratios you need to know the frequencies of the various outcomes in a file of \textbf{linked} pairs of records, and in a corresponding file of \textbf{unlinked} records. However, linkage would not usually have been done at this stage, so how are these frequencies to be obtained? Starting from scratch, therefore, the user will usually have to employ one of the following methods:

1) An initial linkage process of pairs of records from the file (after first sorting and blocking to improve efficiency) could perhaps be carried out manually, by exact matching or by some other means. This linkage does not have to be perfect, nor the numbers of records large, about 1,000 would normally be sufficient.

2) Where a previous linkage exercise has already been undertaken on other files, the outcome frequencies from this process may be used instead, but care must be taken to ensure that the two files are of the same quality, and represent the same population.

3) Frequency ratios for name and address may be determined empirically by a method as described in 1) or 2) above, but frequency ratios for some identifiers like date of birth
can readily be calculated from their known distribution in the population. For example, month of birth has only 12 values and since birthday is approximately uniformly distributed throughout the year, the probability of matching in the absence of errors is approximately 1/12 in truly unlinked pairs, and in linked pairs is 1. The probability of disagreement in truly linked pairs is 0, and in truly unlinked pairs is approximately 11/12. In this way one is usually able to predict what the expected frequency ratio for unlinked pairs will be. So the frequency ratio for agreement that arises from these calculations is 12, and for disagreement is 0. In practice there would usually be some coding or transcription error rates to take account of, so these values would be slightly modified to allow for these errors, and zeroes would not generally occur.

4) Another modification to this procedure is to remove the requirement to calculate the denominator frequency ratios from unlinked pairs: often the relative frequencies in the denominator can, with little loss of accuracy, be calculated using pairs of records drawn at random from the file, rather than specifically unlinked pairs. So the relative frequencies can be calculated from a single file, sometimes called a global file. (An alternative terminology is that the relative frequencies are referred to as population relative frequencies.) Of course this is an approximation, but for many purposes it is adequate.

5) Another source of frequency ratios for name and address identifiers apart from the use of methods (1) and (2) above, using previously published values. Newcombe (1998) gives some typical values that are shown in Box 7.2. The global frequency ratios shown in Box 7.2 do not take into account whether the particular values the identifier takes are common or rare events in the population. The intuition underlying outcome-specific frequency ratios is that surnames such as ZACHARIAS occur less often than surnames such as SMITH. Thus pairs of records both containing the name ZACHARIAS are more likely to indicate a true linkage, than pairs of records both containing the name SMITH.

Outcome specific ratios are not difficult to use in principle, with the following definition

\[
\text{OUTCOME SPECIFIC FREQUENCY RATIO } = \frac{\text{relative frequency of agreement (x,y) among linked pairs}}{\text{relative frequency of agreement (x,y) among unlinked pairs}}
\]

(\text{for agreement on JOHN})

where \(x\) indicates the value of the identifier on the query or data record, and \(y\) indicates the value of the identifier on the candidate or master file record.
Box 7.2 Global frequency ratios for agreements and disagreements of selected identifiers on matched pairs of records

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Comparison outcomes</th>
<th>Relative Frequencies (%)</th>
<th>Global Frequency ratios (links/non-links)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Surname</td>
<td>Agree</td>
<td>96.5 0.1</td>
<td>965/1</td>
</tr>
<tr>
<td></td>
<td>Disagree</td>
<td>3.5 99.9</td>
<td>1/29</td>
</tr>
<tr>
<td>Forename</td>
<td>Agree</td>
<td>79.0 0.9</td>
<td>88/1</td>
</tr>
<tr>
<td></td>
<td>Disagree</td>
<td>21.0 99.1</td>
<td>1/5</td>
</tr>
<tr>
<td>Year of Birth</td>
<td>Agree</td>
<td>77.3 1.1</td>
<td>70/1</td>
</tr>
<tr>
<td></td>
<td>Disagree</td>
<td>22.2 98.9</td>
<td>1/4</td>
</tr>
</tbody>
</table>

Source: Newcombe (1988)

The mathematical basis of such intuitive assessments is really quite simple. The greater the ratio of the linked/unlinked frequencies, the greater will be the mathematical weight attached to any particular kind of agreement. Furthermore, one can simplify the process of their calculation using the proportions of occurrences of the string JOHN, say, on both the global and the matched file and without the need to create a file of unlinked pairs, since the difference between relative frequency among the unlinked pairs and that among all pairs (i.e. the global figure, see Box 7.3) will generally be very slight.

Alternative definitions of outcome specific frequency ratios and the resulting match weights have been used (Newcombe, 1988), but for reasons of simplicity of presentation we shall use the above definition only.

Box 7.3 Frequencies of specific values of identifiers

<table>
<thead>
<tr>
<th>Type of Identifier</th>
<th>Value of the Identifier on the search record</th>
<th>Relative frequency in the file being searched</th>
</tr>
</thead>
<tbody>
<tr>
<td>Present Surname</td>
<td>SMITH</td>
<td>1.22%</td>
</tr>
<tr>
<td></td>
<td>HALL</td>
<td>0.27%</td>
</tr>
<tr>
<td>First forename</td>
<td>JOHN</td>
<td>0.14%</td>
</tr>
<tr>
<td></td>
<td>MARGARET</td>
<td>0.06%</td>
</tr>
<tr>
<td></td>
<td>LEICESTER</td>
<td>0.0000076%</td>
</tr>
<tr>
<td>Month of Birth</td>
<td>MARCH</td>
<td>31/365 = 8.49%</td>
</tr>
</tbody>
</table>

Source: ORLS, 1997

7.3 From frequency ratios to match weights

The various centres that have undertaken record linkage on a large scale have invested substantial development resources in the methods of generating weights. These weights are a measure of the log likelihood-ratio that pairs of records, containing arrays of partial identifiers
that may be subject to error or variation in recording, do or do not belong to the same person. Decisions can then be made about the overall level of this ratio to accept or reject the pair for linkage. In deriving and using these weights, we are attempting to, i) reduce the probability of not matching the records that should be matched together, and ii) reduce the probability of matching together those records which should not be matched - see Type 1 and Type 2 errors in Section 7.12. (Winkler, 1995; Scheuren and Winkler, 1996; Holmes, 1975; Jaro, 1995; Gill, 1997).

*Single variable Binit weights*

So far, we have determined frequency ratios for particular identifying variables that give us some idea whether those variables are useful for matching. To summarise those findings, if the frequency ratio for agreement is very low, then the variable is of little use. If it is very high then agreement on that variable is likely to be strongly indicative of a true match. Thus from the figures in Box 7.3 we see that agreement of surname is much more useful in matching than agreement on month of birth. We also see that when the agreement on surname occurs with a rare surname, then the agreement will be even more powerful for matching.

In practice, the frequency ratios are normally transformed using logarithms. The reason for taking logarithms is that multiplying frequency ratios (which we shall see below is necessary when we come to combine the evidence for linkage of a specific record pair over several variables) corresponds to addition of the weights. Following the practice used in information theory, logarithms to the base 2 are used and these are called *Binit weights* that are simply:

\[
W = \log_2 (L/U), \text{ where}
\]

L= relative frequency of agreement/ disagreement among linked pairs

U= relative frequency of agreement/ disagreement among unlinked pairs

The *Binit weights* calculated using \( \log_2 \) are generated from standard \( \log_{10} \) tables in the following way:

\[
\log_2 (L/U) = \frac{\log_{10} (L/U)}{\log_{10} 2} = \frac{\log_{10} (L/U)}{0.30103}
\]

Some typical values for these weights are shown in Box 7.4. Another way of thinking about them is as the power of 2 that equals the frequency ratio. In general, *Binit weights* for agreements will have positive values, and for disagreements the weights will be negative.
### Box 7.4 Binit weights for some common identifiers

<table>
<thead>
<tr>
<th>Agreements or disagreements</th>
<th>Relative frequency in linked pairs (L)</th>
<th>Relative frequency in unlinked pairs (U)</th>
<th>Ratio L/U</th>
<th>Binit weight log2 L/U</th>
</tr>
</thead>
<tbody>
<tr>
<td>Male Sex</td>
<td>1/2</td>
<td>1/4</td>
<td>2</td>
<td>+1</td>
</tr>
<tr>
<td>Initial ‘J’</td>
<td>1/16</td>
<td>1/256</td>
<td>16</td>
<td>+4</td>
</tr>
<tr>
<td>Initial ‘Z’</td>
<td>1/1000</td>
<td>1/1000000</td>
<td>1000</td>
<td>+10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Disagreements</th>
<th>Relative frequency in linked pairs (L)</th>
<th>Relative frequency in unlinked pairs (U)</th>
<th>Ratio L/U</th>
<th>Binit weight log2 L/U</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sex</td>
<td>1/8000</td>
<td>1/2</td>
<td>1/4000</td>
<td>-12</td>
</tr>
<tr>
<td>Initial ‘J’</td>
<td>1/40</td>
<td>32/40</td>
<td>1/32</td>
<td>-5</td>
</tr>
</tbody>
</table>

Source: Newcombe (1988)

When a variable is subject to coding or transcription error, the L relative frequency or probability is effectively one minus the error rate. The more reliable the measurement of the variable, the closer to 1 the estimated probability will be. The U probability is the probability that a variable agrees given that the record pair is for different people selected at random. Since there are many more unlinked pairs than linked pairs this probability is, in most cases, effectively the probability that the values of the variable agree at random.

### 7.4 Combination of weights over all variables in the identifying set

The question now arises: how do we use the frequency ratios (and their equivalents – the single variable Binit weights) in a practical matching exercise. What we need to do is to obtain the equivalent of a single frequency ratio for all the variables in the identifying set that are to be used in our matching exercise. We first explain what we do with them, and after that the rationale behind the procedure in more detail. Unsurprisingly, the overall match weights are calculated from frequency ratios. We multiply together the frequency ratios for the agreements and disagreements that actually occur in each field. Suppose that the frequency ratios in each of the three variables used in a particular matching exercise are \( f_{a1}, f_{a2}, f_{a3} \) and \( f_{d1}, f_{d2}, f_{d3} \) respectively for agreement and disagreement, and, for the particular record pair being evaluated, there was agreement on variable 1, disagreement on variable 2, and agreement on variable 3. Then we would multiply together \( f_{a1}f_{d2}f_{a3} \) to obtain an overall frequency ratio or score, \( f_{a1}f_{d2}f_{a3} \), for what was observed in the specific record pair. This score can be interpreted as the likelihood of true linkage relative to the likelihood of non-linkage for this record pair based on this evidence. For example, the variables could be surname, sex and year of birth, then \( f_{a1} = 965 \) (see Box 7.2), \( f_{d2} = 1/4000 \) (Box 7.4), \( f_{a3} = 70 \) (Box 7.2), the overall relative frequency would be \( 965 \times (1/4000) \times 70 = 16.9 \). So from this evidence, despite the fact that there was a disagreement on sex, a linkage is more likely than a non-linkage.

This calculation of the overall frequency ratio is based on the assumption that there is no correlation between the variables that constitute the fields. This is probably true for some variables, e.g. month of birth which is usually independent of forename. However some forenames and surnames are likely to be correlated in some populations. For example in the population in the UK, forenames and surnames which are characteristic of ethnic groups or particular nationalities are more likely to occur together: Marcel is more likely to go with Duchamp than Smith. However, in most practical matching exercises, this complication is ignored, and the frequency ratios are multiplied together. One solution to the problem of correlated variables is to use the combined variable as a single variable in the calculation of
frequency ratios. Another example occurs in a medical context in the case of a change of address and hospital code. If a family moves their home for example, one might expect that variables such as house number, street, town, family doctor and hospital code to change all at once. The frequency ratio can again be calculated considering all the variables as a single unit.

Of course, multiplying the frequency ratios corresponds to adding together the Binit weights. The information contained in every matching variable, when this is combined together over all variables by adding the Binit weights, helps to determine which records should be identified as linked and which not. Each variable generally provides a part of the information but some variables (e.g. NHS number) provide more information than others (e.g. sex). Nevertheless, it is usually the case that, taken together, the weights for all the variables - especially when they do not contain anything quite so identifying on its own as an NHS number - would normally determine whether the two records match or not much better than any individual variable alone.

The total Binit weight now indicates, on a logarithmic scale, the ratio of the likelihood that two records should be linked versus the likelihood that they should not be linked based on the information from all the variables in the identifying set (Smith, 1984; Howe and Lindsay, 1981). A total weight of zero represents a relative likelihood of 1:1 that the linkage is a correct one, each added weight unit doubling the relative likelihood and each subtracted unit halving it. For example, weights of +1 and +2 represent likelihood ratios of 2:1 and 4:1, respectively, in favour of a correct match, whereas weights of -1 and -2 represent likelihood ratios of 1:2 and 1:4 and so argue against a correct match. The weights represent potential links in decreasing order of certainty (Howe and Lindsay, 1981; Newcombe et al., 1983).

An example: matching on month and day of birth

To clarify this process, let us consider a simple example in which we are matching on month of birth in two records that belong to the same person. The variable is available 100 per cent of the time and any differences will be due to coding or transcription errors and let us assume that the error is about three per cent. The L probability will then be:

\[ L = (1 - \text{error}) = (1 - 0.03) = 0.97 \]

and therefore the fields for month of birth for any given matched record pair will match 97 per cent of the time. The U probability is the frequency of agreements of two records that do not belong to the same person, and since there are 12 months in a year, the random agreement U will occur 1/12 or 8.3 per cent of the time. The likelihood ratio of two records agreeing on month of birth will then be \( 0.97 / 0.083 = 11.64 \). This means that agreement on the month of birth alone increases the likelihood that the two records belong to the same person by a factor of 11.64.

For each record pair being compared, a composite weight may be computed as the sum of the individual Binit (\( \log_2 \)) weights. Where the same variable agrees on each of the pair of records, the outcome specific weights are computed as above. For a variable that disagrees in each of the pair of records being matched, the disagreement weight will be computed as:

\[ \log_2 \left( \frac{1 - L}{1 - U} \right) \]

Therefore, the disagreement weight for month of birth will be,

\[ \log_2 \left( \frac{1 - 0.97}{1 - 0.0833} \right) = \log_2 [0.03/0.917] = \log_2 [0.0327] = -4.93 \]
If the same calculation is carried out for the day of birth (assume an error rate of five per cent and an average of 30 days in each month) the agreement weight will be

\[ = \log_2 \left( \frac{1 - 0.05}{0.03333} \right) = \log_2 (28.50) = +4.83 \]

and the disagreement weight will be

\[ = \log_2 \left( \frac{0.05}{1.0 - 0.03333} \right) = \log_2 (0.0517) = -4.27 \]

Finally, if it is assumed that the month-of-birth and day-of-birth are independently distributed in the population and that reporting errors for matched pairs are independent, with these stipulations and assumptions, we can calculate the probabilities as shown in Box 7.5. Usually, given the independence assumption, the probability ratio is broken up into a series of ratios, one for each agreement or disagreement, and logarithms are taken (to the base 2 in this example). One is now working with simple sums, such that the larger (more positive) the total, the more likely that the pair is a match; conversely, the more negative the sum, the greater the likelihood that the two records are not for the same person.

<table>
<thead>
<tr>
<th>Box 7.5 Probability ratios for outcomes based on day-of-birth and month-of-birth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outcome of comparison</td>
</tr>
<tr>
<td>Month of Birth (MOB) agrees</td>
</tr>
<tr>
<td>Month of Birth disagrees</td>
</tr>
<tr>
<td>Day of Birth (DOB) agrees</td>
</tr>
<tr>
<td>Day of Birth disagrees</td>
</tr>
<tr>
<td>MOB and DOB agree</td>
</tr>
<tr>
<td>MOB agrees / DOB disagrees</td>
</tr>
<tr>
<td>MOB disagrees / DOB agrees</td>
</tr>
<tr>
<td>Neither agrees</td>
</tr>
</tbody>
</table>

In this example it is only when both day-of-birth and month-of-birth agree that the sum of the logarithms is highly positive (+8.38 obtained from 3.54 + 4.83). As one would expect, the strongest evidence in favour of a non-match occurs when both day-of-birth and month-of-birth do not agree; for this outcome the \textit{Binit} value of the probability is about -9.21. This example illustrates nicely the fact that outcomes that are frequent in the population do not add very much to one's ability to decide if the pair should be treated as linked. However, if there are disagreements on such variables and the reporting is reasonably accurate, then a combination of the variables may have a great deal of power in identifying comparison pairs that represent non-links.

### 7.5 Generation of outcome specific weights

There is a further refinement about the weights which corresponds to the distinction we made about frequency ratios. There are two types of probabilistic matching that differ according to how the logarithmic weights are generated. A linkage with \textbf{global weights}, as described above, generates the outcome weights based on whether a given identifier agrees or disagrees
with its counterpart on the other file. Probabilistic linkage with **outcome-specific weights** resolves matches more accurately since although it is often based on the proportion of the links and non-links in the file, it involves more intricate and sophisticated weight calculation tailored to the specific pair of outcomes under consideration, as we discussed earlier when defining outcome specific frequency ratios. Each outcome among several is assigned a weight.

In Section 7.2, a formula is given for outcome specific frequency ratios which can be used only when the results of a prior matching exercise are available. To start with, in many applications we do not have prior information on the population of linked and non-linked records, nor the proportions of agreement and disagreement, so these weights can not be calculated. Instead, approximate outcome specific weights are generated. A good approximation to the outcome specific weights (in units of \( \log_2 \), or \( \text{Binit} \)) for agreements is given by:

\[
W = \log_2 \left( \frac{1}{p_i} \right), \text{ where } p_i \text{ is the proportion of the given outcome in the population.}
\]

In particular note that the approximate outcome specific weights are based just on the proportions of the particular outcome in the population. This procedure assumes that the variables are measured without error (the numerator is 1) and that the global frequency (ignoring linkage status) is a good approximation to the probability of the outcome assuming no link.

In some situations, the tables of proportions can be created **on-the-fly** using the files actually being matched (Winkler, 1989c) while in others the tables of proportions are created a **priori** using large reference files. The advantage of **on-the-fly** tables is that they can use different relative proportions in different geographic regions; for example, matching records with Asian surnames in Yorkshire and Leicestershire, or Arabic names in London. The disadvantage of **on-the-fly** tables is that they must be based on files that cover a large percentage of the target population. If the data files contain samples from a population, then the outcome specific weights should reflect the proportions of the appropriate populations.

**Characteristics of variables for record matching purposes**

Before discussing complications associated with the various types of outcome-specific weights (forename, surname, etc.), we present in Box 7.6 a table showing the use and availability of each variable in terms of: discrimination, liability to change over a person's lifetime, liability to error and their availability in routinely collected data.

**Calculation of outcome specific weights for surnames.**

The ORLS found that the outcome specific weights calculated from the frequency of the first letter in the surname (26 different values) was too crude for matching together files that contained over 1 million records. The weights for SMITH, SNAITH, SNEATH, SMOOTHEY, SAMUDA and SZABO (which all fall into the same ONCA block S530) would all be set to some low value calculated from the frequency of the initial letter 'S' in the population, and would be based on the frequency of SMITH and ignoring the frequency of the much rarer SAMUDA.

Using the frequencies of all of the 1 million or so different surnames on the master match file is too cumbersome, too time consuming to keep up-to-date, and operationally difficult to store and access during a match run. The list would also contain most of the one-off surnames generated by poor transcription and bad spelling. A compromise solution was devised by calculating the **Binit weights** based on the frequency of the ONCA block (roughly 8000 values), with a cut-off value of 1 in a 100,000 in order to prevent the very rare or one-off names from carrying very high weights (see Box 7.7). Although this approach does not get
round the problem of the very different names that can be found in the same ONCA block, it
does provide a higher level of discrimination and can to some extent accommodate the low
frequency or erroneous names.

Box 7.6 Relative characteristics of identifying variables used for person matching in
health-related datasets.

<table>
<thead>
<tr>
<th>Identifying variable</th>
<th>Characteristics</th>
<th>Notes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Present Surname</td>
<td>Av Av Hi Hi</td>
<td>Always available</td>
</tr>
<tr>
<td>Birth Surname</td>
<td>Av Hi Hi Lo</td>
<td>Limited availability</td>
</tr>
<tr>
<td>Forenames</td>
<td>Av Hi Hi Hi</td>
<td>Use at least two forenames</td>
</tr>
<tr>
<td>Date of Birth</td>
<td>Av Hi Hi Hi</td>
<td>Should be checked against age</td>
</tr>
<tr>
<td>Place of Birth</td>
<td>Av Hi Av Lo</td>
<td>Parent's residence at birth</td>
</tr>
<tr>
<td>Gender (Sex)</td>
<td>Lo Hi Lo Hi</td>
<td>Only variable that is unambiguous</td>
</tr>
<tr>
<td>Marital Status</td>
<td>Av Lo Av Lo</td>
<td>Useful for family linking</td>
</tr>
<tr>
<td>Usual address</td>
<td>Av Lo Av Hi</td>
<td>Full postal address</td>
</tr>
<tr>
<td>Postcode</td>
<td>Av Lo Av Hi</td>
<td>Full postcode</td>
</tr>
<tr>
<td>NHS Number</td>
<td>Hi Hi Hi Av</td>
<td>Must not be recorded from memory</td>
</tr>
<tr>
<td>GP</td>
<td>Av Lo Av Hi</td>
<td>Useful for checking address</td>
</tr>
<tr>
<td>Date of marriage</td>
<td>Av Hi Av Lo</td>
<td>For family record linkage</td>
</tr>
<tr>
<td>Mother's Birth Surname</td>
<td>Av Hi Av Lo</td>
<td>For family record linkage</td>
</tr>
<tr>
<td>Hospital Unit Number</td>
<td>Av Hi Av Av</td>
<td>Valuable if has check character</td>
</tr>
<tr>
<td>Hospital site code</td>
<td>Av Hi Av Av</td>
<td>Used with Hospital Unit number</td>
</tr>
</tbody>
</table>

Notes:
- Dp = Discriminating power
- St = Stability, i.e. liability to change
- Err = Liability to error
- Cp = Availability, results of capturing this variable
- Hi = High
- Av = Average
- Lo = Low

Source: (Baldwin, 1972; Gill & Baldwin, 1987)

When the two names variables are being compared during the matching run, one from the
data file and the other from the master file, a string comparison algorithm is used to compute
a weight modification factor (N) based on the frequency of the surname (see Boxes 7.11,
7.12, 7.13) according to an algorithm devised by Knuth-Morris-Pratt (Stephen, 1994; Gonnet
and Baeza-Yates, 1991). (see Section 7.5) This algorithm computes the amount of agreement
and disagreement in the two name strings, and uses: the length of the shortest of the two
names being compared, the difference in length of the two names, the number of letters
agreeing, the number of letters disagreeing and the number of character transpositions to
prepare a names weight modification factor.
### Box 7.7 Calculation of outcome specific weight for a surname

<table>
<thead>
<tr>
<th>Frequency of TAYLOR</th>
<th>$\frac{326,669}{57,963,992} = 0.0056357$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outcome specific weight</td>
<td>$\log_2 (1 / 0.0056357) = \log_2 (177.439) = 7.47$</td>
</tr>
<tr>
<td>Frequency of SZABO</td>
<td>$\frac{329}{57,963,992} = 0.000005675$</td>
</tr>
<tr>
<td>Outcome specific weight</td>
<td>$\log_2 (176,182.347) = 17.43$</td>
</tr>
<tr>
<td>Truncated outcome specific weight</td>
<td>$= 17.0$</td>
</tr>
</tbody>
</table>

Source: NHSCR (1997)

The product of the agreement weight and the above factor would yield a modified weight that reflects the outcome specific weight for the two names, multiplied by the amount of agreement between the two names. Where the two name strings are absolutely identical, the weight will be computed as $+2N$ (N is the Binit weight), but would be reduced to a lower value of $-2N$ where the amount of disagreement is quite large, or the names are dissimilar.

In cases where the birth surname and present surname are swapped with each other, or other surnames are recorded, expanding the file (as described in Section 5.2) would enable the location of, and access to, the blocks that contain the records stored under the different versions of the surnames. The specific outcome weight for the PSN/PSN (PSN=Present surname) and the BSN/BSN (BSN=Birth surname) pairings are first calculated, then the PSN/BSN and the BSN/PSN pairings are also calculated. The higher of the two values is used in the subsequent calculations for the derivation of the outcome specific weight.

Where a person has a marital status recorded as single, i.e. never married; or the sex is male; or the sex is female and the age is less that 16 years, it is normal practice in the UK for the present surname to be the same as the birth surname, and for this reason only the weight for the present surname will be calculated and used for the determination of a match. It is assumed that where the birth surname is the same as the present surname, no new information is being recorded.

### Calculation of outcome specific weights for forenames

The weights derived for the forenames could be based on the frequency of the initial letter of the forename in the population, but it is recommended that the weight is generated in a fashion similar to that described for surnames, above. The distribution of male and female forenames are different, so there would be two different sets of weights, one for males and a second for females. Since the forenames can be recorded in any order, for example, FN1 and FN2 in the first record and FN2 and FN1 in the second record, the weights for all the forename combinations are calculated and the highest values used for the match. Another method of weighting and comparing forenames would be to sort them into alphabetic order.

Where there are wide variations in the spelling of the forenames, the ORLS are evaluating the use of the Daitch-Motokoff version of Soundex for phonetically compressing and weighting the forenames in a fashion similar to that used for the surnames.
Box 7.8 Calculation of outcome specific weight for a forename

\[
\text{Frequency of SARAH} \quad 24,951 / 28,989,996 = 0.0008606762 \\
\text{Outcome specific weight} \quad = \log_2 (1 / 0.00086067) = \log_2 (1161.877) \\
\quad = 10.18
\]

\[
\text{Frequency of LEICESTER} \quad 44 / 28,989,996 = 0.0000015178 \\
\text{Outcome specific weight} \quad = \log_2 (1 / 0.0000015178) \\
\quad = 19.33
\]

Source: NHSCR (1997)

Calculation of outcome specific weights for non-names variables

The outcome specific weights for date of birth, sex, place of birth and NHS number are calculated using the frequency of the variable in the population. The weight for the year-of-birth comparison can been extended to allow for expected errors, for example only a small deduction should be made where the two date of births differ by exactly 1 or 10 years, but reduced substantially where the two date of births differ by, say, 7 years.

In the ORLS system, the weight for the street address is based on the first 8 characters of the full street address, where these characters signify a house number (31, High Street), or house name (High Trees), or a public house name (THE RED LION), or organisational addresses. In parsing the address, terms like 'Flat' or 'Apartment' can be ignored and other parts of the address then used for the comparison. The weights for house numbers are based on short streets and low numbers, the weights for house names are based on the total proportion of house names in the population.

The postcode is treated and weighted as a single variable although the inward and outward parts of the code could be weighted and used separately.

Limiting the values of the outcome specific weights for very rare or unusual names

In the calculation of outcome specific weights for rare or unusual names, the value of the weight needs to be restricted to prevent records being matched together irrespective of any other information that is present in the matched pair.

The outcome specific weights for the non names variables are generated from the distribution of the variable in the population and with the exception of house name, are evenly distributed. This is not the case for names information where the weights can be quite low for common names like SMITH or JOHN or very high for ZACHARIUS or ZZDINSKY. To prevent situations with very high weights from exceeding the threshold value, the following approach has been adopted by the ORLS:

i) to impose a ceiling on the value of the outcome weight for rare names to an outcome weight of, say, 17, that is a probability of about 1 in 100,000, and,

ii) to use dual thresholds, that is to consider the combined outcome weight for names and non-names quite separately (see Section 7.9).
Box 7.9 Calculation of outcome specific weight for a date of birth

<table>
<thead>
<tr>
<th>Frequency of DAY of birth</th>
<th>1/31</th>
<th>Outcome weight (= \log_2 \left( \frac{1}{1/31} \right) = \log_2 (31) = 4.95 ) usual rounded to 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency of MONTH of birth</td>
<td>1/12</td>
<td>Outcome weight (= \log_2 \left( \frac{1}{1/12} \right) = \log_2 (12) = 3.58 ) usual rounded to 4</td>
</tr>
<tr>
<td>Frequency of YEAR of birth</td>
<td>1/70</td>
<td>Outcome weight (= \log_2 \left( \frac{1}{1/70} \right) = \log_2 (70) = 6.12 ) usual rounded to 6</td>
</tr>
<tr>
<td>Outcome weight for the full Date of Birth</td>
<td>5+4+6 = 15</td>
<td></td>
</tr>
</tbody>
</table>

Source: ORLS

Calculation of outcome specific weights where the two variables do not quite agree or where the two variables disagree completely

The string matching algorithms are used to detect and quantify the differences between the two text strings. The outcome specific weight will be positive for two similar name strings and negative for two dissimilar strings, for example:

- G600 GRAY / GRAY outcome specific weight = +19.0
- G610 GREAVES / GRAVES outcome specific weight = +15.5
- G620 GRACE / GEORGE outcome specific weight = -2.0
- S530 SMITH / SZABO outcome specific weight = -12.0

The weights have been calculated using the differences between the data and master files and using these proportions to calculate the outcome specific weights. Some weights can be modified to take account of the more or less common types of error. For example, an erroneous year of birth is usually 1 or 10 years discrepant but very rarely 3 years discrepant.

Examples of the match between two successive dates of birth for the same person on the ORLS file are shown below:

- 16041957 / 16041957 outcome specific weight = +15 (same date)
- 29031996 / 09031995 outcome specific weight = +3 (20 days/1 year discrepant)
- 11121907 / 16041908 outcome specific weight = -5 (5 days/8 months/1 year)
- 15102000 / 13041967 outcome specific weight = -15 (all different)

The calculated weight can become negative where there is extreme disagreement between the variable on the data record and the corresponding variable on the master file. In matching street address, postcode and general practitioner, for example, the weight cannot become negative, although it can be zero, because the person may have changed their home address or their family doctor since they were last entered into the system, this is really a change in family circumstances and are not errors in the data and so a negative weight is not justified.

The table presented in Box 7.10 shows the range of outcome specific weights that are typically used in the ORLS system. For each of the identifying variables the range of outcome specific weights are shown. For example, the match on exactly the same surnames would generate outcome weights in the range \(2^6 = 12\) up to \(2^{17} = 34\) depending on the proportion of the surname in the population, i.e. SMITH would be \(2^6 = 12\), and ZACHARIAS would be \(2^{17} = 34\). Likewise the forenames would be in the range \(2^9 = 18\) up to \(2^{20} = 40\). In similar fashion, comparison of two dates of birth would produce a weight of +15 if the two dates were exactly the same, and in the range +15 to -15 if dates are different. Where one date of
birth is set to some arbitrary value (for example, 01011852 for not known dates) the outcome weight would be set to -15.

The outcome specific weights shown in Box 7.10 have been developed and used over the 38 years experience of the ORLS.

**Box 7.10 The outcome specific weights used by ORLS for matching**

<table>
<thead>
<tr>
<th>Identifying Variable</th>
<th>Outcome Weight (see note #1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Surnames:</td>
<td></td>
</tr>
<tr>
<td>Birth surname</td>
<td>+2S</td>
</tr>
<tr>
<td>Present surname (note #2)</td>
<td>+2S to -2S</td>
</tr>
<tr>
<td>Mother's birth surname</td>
<td>+2S to -2S</td>
</tr>
<tr>
<td></td>
<td>(where: common surname S = 6, rare surname S = 17)</td>
</tr>
<tr>
<td>Forenames (note #3)</td>
<td>+2F</td>
</tr>
<tr>
<td></td>
<td>(where: common forename F = 9, rare forename F = 20)</td>
</tr>
<tr>
<td>NHS number</td>
<td>+10</td>
</tr>
<tr>
<td>Place of birth (code)</td>
<td>+4</td>
</tr>
<tr>
<td>Street address (note #5)</td>
<td>+5, +7</td>
</tr>
<tr>
<td>Post Code</td>
<td>+4</td>
</tr>
<tr>
<td>GP (code)</td>
<td>+4</td>
</tr>
<tr>
<td>Sex (note #6)</td>
<td>+1</td>
</tr>
<tr>
<td>Date of birth</td>
<td>+15</td>
</tr>
<tr>
<td>Hospital and</td>
<td></td>
</tr>
<tr>
<td>Hospital unit number</td>
<td>+7</td>
</tr>
</tbody>
</table>

Notes:

1. Where an variable has been recorded as not known, the variable has been left blank, or filled with an error flag, the match weight will be set to 0, except for special values described in the following notes.
2. Where the surname is not known or has been entered as blank, the record cannot be matched in the usual way, but it is added to the file to enable true counts of all the events to be made.
3. Forename entries, such as boy, girl, baby, infant, twin, or not known, are weighted as -10.
4. Where the weight is shown as NP (not permissible), this partially known value cannot be weighted in the normal fashion and is treated as a NO MATCH.
5. No fixed abode is scored 0.
6. Where sex is not known, blank, or in error, it is scored -10. (All records input to the match are checked against forename/sex indexes and the sex is set to M or F where it is missing or in error).

Source: (Gill, 1997)

**String comparison methods and their use for the calculation of outcome specific weights in names**

In this Section we examine the algorithms for text searching in strings. It is a basic part of indexing and also used for comparing strings for pattern matching. The comparison of two character or text strings is a classical problem to which a wealth of solutions exists, and the rest of this Section will cover the main algorithms used in record matching and will not include any of the modern theoretical work.
Dealing with typographical error is crucially important in record matching and linkage. If the strings can only be compared on an exact character by character basis, many potential matches will be lost. The Post Enumeration Survey (PES) (Winkler and Thibaudeau, 1991; Jaro, 1989) reported that among the many true matches, almost 20 per cent of surnames and 25 per cent of forenames disagreed. The ORLS found that only 75-85 per cent of the matched records exactly matched on surnames alone. If the matching had been done on this basis, more than 30 per cent of the matches would have been missed by the computer algorithms, resulting on the one hand in substantially increased clerical checking, or on the other hand a poorly matched file.

Where there are discrepancies between the two texts, how can the error be measured? One method of solution uses the Levenstein distance which is a distance measure between two text strings, determined by the maximum number of insertions, deletions and substitutions required to transform one string into another string. With minimal modifications the algorithm can be adapted to searching whole words matching with k errors.

The string comparison algorithm first computes the number of insertions, deletions, transpositions, and the length of the string. The algorithm then uses this information, i.e. the length of the shortest of the two names being compared, the difference in length of the two names, the number of letters agreeing and the number of letters disagreeing to calculate an agreement factor with which to generate the outcome weight. During the comparison, adjacent pairs of characters are switched around to test for transpositions that might occur in either of the strings.

The string comparison methods are also used in the calculation of the outcome specific weights between two name strings. The final outcome weight in the comparison of two names is based on the theoretical weight calculated from the frequency of the surnames in the population and modified using a method based on the Knuth-Morris-Pratt (KMP) algorithm (Gill, 1997; Stephen, 1994; Gonnet and Baeza-Yates, 1991, 1999; Crochemore and Rytter, 1994).

**Brute force string comparison methods**

The brute-force (BF) algorithm is the simplest one to use. It consists merely of trying all possible pattern positions in the text string. For each such position, it verifies whether the pattern matches at that position (Baeza Yates, 1999).

Using the first string as the pattern, match the other string against it. If it matches, fine, if not then move the pattern along by one character and try again. While this brute force technique will work, it involves an unnecessarily large amount of work. If one string contains m characters and the other string n characters, then in the worst case m*n comparisons are necessary. Using many different types of string matching, it is easy show that the brute force technique requires so much work because it may repeat many comparisons. There is no memory of the comparisons that have been made. If there were some method of recording which comparisons had been made, the algorithm could be altered to omit these comparisons on subsequent occasions. The following techniques attempt to do this. Whenever a character in the pattern string is compared with a character in the text string, one of two things can happen: either they match or they do not. If they match, the examination continues with the next character. However, if they do not match, the character that was encountered in the text string is known. The solution is then to store this information and to relate it to succeeding comparisons.

**The Knuth-Morris-Pratt (KMP) algorithm**

The Knuth-Morris-Pratt known as the KMP algorithm (Knuth, Morris and Pratt, 1977; Baeza Yates, 1999), was the first with linear worst-case behaviour, although it is not much faster
than the Brute Force algorithm. This algorithm slides a window over the text. It does not try all the window positions as does the Brute Force method, instead it reuses information from the previous checks. The KMP algorithm is built around the concept of storing the initial sequences of the pattern string. A modification of the KMP algorithm is used for the comparison of the strings in the ORLS system, (Gill, 1997) and in the Winkler and Jaro systems (Jaro, 1989; Winkler, 1990b).

Boyer-Moore (BM) algorithm

Boyer-Moore algorithms although similar in concept to the KMP are based on the fact that the check inside the window can proceed backwards as well as forwards.

The ORLS string comparison method

The comparison algorithm used for the ORLS matching is presented in Box 7.11. The method is based on counting the number of characters in the two name strings that agree, the number that disagree, the number of character pairs that are transposed, the length of the longest string, and the length of the shortest string. The final outcome specific weight is computed from the initial weight determined from the frequency of the name in the population multiplied by the modification factor.

The algorithm has been refined to give extra weight to the characters that agree at the beginning of the string. Where the two name strings are absolutely identical, the weight approaches +2N, (where N is the theoretical initial outcome specific weight based on the value of the variable) but falls down to a lower value of -2N where the amount of disagreement is quite large.

Winkler-Jaro algorithm

Jaro (1989) introduced methods for dealing with typographical error such as 'SMITH' versus 'SMOTH'. Jaro's procedure consists of two steps. First, a string comparator returns a value based on counting insertions, deletions, transpositions, and string length. Second, the value is used to adjust a total outcome weight downward toward the total disagreement weight. Jaro's string comparator was extended by making agreement in the first few characters of the string more important than agreement in the last few (Winkler, 1990b). The original Jaro comparator and the Winkler-enhanced comparator yield a more refined scale for describing the effects of typographical error than do standard computer science methods such as the Damerau-Levenshtein metric (Winkler, 1985a, 1990b). The basic Jaro algorithm computes the string lengths, finds the number of common characters in the two strings, and finds the number of transpositions, and is presented in Box 7.12. The Jaro algorithm has been enhanced by McLaughlin (1993), Winkler (1993) and Lynch and Winkler (1994).

Use of Bigrams for string comparisons

Another common method of comparing two strings is by comparing the Bigrams that two strings have in common, as presented in Box 7.13. The original algorithm cannot compensate for character transpositions in the strings although a modification of the algorithm is to switch the pairs of characters around in each bigram.

The Bigram algorithm returns a value between 0 and 1. Bigrams are known to be very effective in dealing with minor typographical errors and Porter and Winkler have shown empirically that Bigrams work well (Porter & Winkler, 1999; Frakes & Baeza-Yates, 1992).
Box 7.11 ORLS String Comparison method

\[
\text{Final weight} = \text{initial weight} + \{ \text{initial weight} / \text{lenmin} \times (\text{#agree} - 3\#\text{disagree} - 1\#\text{diff} - 1\#\text{transposition})\}
\]

where:
- Final weight is the computed weight taking into account all the differences between the two strings
- Initial weight is the theoretical weight based on the frequency of the identifier in the population
- lenmin length of the shortest string
- agree number of characters that agree
- disagree number of characters that disagree
- transpositions number of two-character transpositions

Box 7.12 Winkler and Jaro string comparator

\[
\text{Jaro}(s1,s2) = \frac{1}{3}(\text{#common/length1} + \text{#common/length2} + 0.5\times\text{#transpositions/#common})
\]

Box 7.13 Comparison of two strings using Bigrams

| LEICESTER | LE EI IC CE ES ST TE ER |
| LESTER | LE ES ST TE ER |

comparison weight = number of bigrams in common / average number of bigrams

\[
= \frac{5}{6.5} = 0.77
\]

7.6 Blocking the file to reduce the number of unproductive matches

As previously described in Section 5.3, all possible combinations of the record pairs on both files need to be tested against each other. Since the number of comparisons using this method is very large, in practice one or more of the matching variables are used to block the files. The practice of blocking provides a practical method of limiting the number of pairs that have to be examined. If both the data and master files are blocked into mutually exclusive subsets and only matches within the subsets are undertaken the whole process becomes manageable, economic and capable of being completed in a reasonable time (see Section 5.3). This results in a file which is divided into smaller blocks in which the candidate record pairs can be compared in a more efficient and less time consuming way. One important consideration is that the number of records in each block should be small enough to avoid many unproductive comparisons and yet large enough to prevent records for the same person spilling over into different blocks and so failing to be compared. The best blocking variables are those with the highest number of values, the highest reliability and stability and the lowest error rates. In similar fashion those variables with a high probability of error should be avoided, for example, street address. In mathematical terms, the variables with the highest weights make the best blocking variables.
7.7 File blocking and matching where the variables used for the blocking keys to exhibit some error

All name search and record matching systems exhibit a conflict between performance and reliability. The choice of blocking keys creates a conflict between the accuracy and performance of the match. Any key that improves the performance but is not reliable cannot be used and vice versa. File blocking should always be organised using the most reliable and accurate variable that is in the data set, providing that the variable has enough discriminating power to divide the file into small but manageable blocks. For example, gender is a very reliable variable but it only has enough discriminating power to divide the file into two parts. The year of birth might be quite useless if it contains a high proportion of errors, and the phonetic code of the surname would divide the file into roughly 8000 blocks.

Many of the file blocking keys selected from a record will exhibit a small amount of error or omission, for example the date of birth may be a few days different, or one month different or one year different. In using geographical identifiers, the postcode may have the terminal character missing, or the street address could have a spelling error. The portion of the address that is the most reliable and the most discriminatory is the dwelling or business number and the street address. The full street address has the most discrimination, but it also has the most error and poor stability since people move around the country. The postcode which is derived from the street address will also suffer from this lack of stability. One way of managing the error in the date of birth is to use the derived age, or age group, or a year of birth group. The benefit of using the year of birth rather than age is that the year of birth is fixed for all time, whereas the age will change with each successive year.

It is often necessary to match two files together using dates for the blocking keys which contain some small errors, for example date of delivery and date of birth (up to 3 days discrepant) or date of discharge from care and date of death (up to 3 days discrepant). In these cases the files need to be blocked in overlapping blocks since the date on the first record could be one side of the true value and the date on the second record on the other side of the true value. In this example, the records would fall into adjacent blocks. The solution will be to store an exact copy of the records in the appropriate block together with a similar copy of the record in the adjacent blocks. This blocking method would enlarge the size of the file (analogous to the file expansion in Section 5.2) but the probability of matching the two files together will be substantially increased.

Yet another way of blocking the file would be to generate the Julian Date (number of elapsed days from some arbitrary start date) from the date and generate extra records for the true date and two or three days either side. This technique has been used quite successfully for matching hospital records for young babies with their other hospital and registration records.

7.8 Constraints on matching to reduce the number of unproductive comparisons

To reduce the number of unproductive comparisons, a data record should only be matched with a record in the corresponding master file block provided that pre-defined rules are satisfied. For example, in the ORLS system, records are only matched where the year of birth on both records is within 16 years of each other. This constraint is applied, firstly, to reduce the number of unproductive matches, and secondly to restrict matching to those persons born within the same generation, and in this way eliminate Father/Son, Mother/ Daughter matches. Further constraints could be built into the matching software to limit unproductive matches, for example, matching only within the same sex, logically checking that the dates on the two records are in a particular sequence or range, or that the diagnoses on the two records are in a specified range, such as that required for the preparation of a disease registry file.
7.9 Determination of the match acceptance threshold

The matching phase in an automatic record matching and linkage system normally consists of the three stages:

i) matching the record pairs (see Section 7.9.1),
ii) setting the matching threshold (see Section 7.10), and
iii) resolution of uncertainties (see Section 7.11).

In probabilistic systems the preliminary decisions are based on the outcome weights assigned to the observed outcomes of the comparisons; in ad hoc systems these decisions are usually based on predetermined rules relating comparison outcomes to the possible linkage decisions. Most automatic systems require manual intervention at some stage, especially to resolve uncertainties that remain after the preliminary linkage decision step has been taken.

7.9.1 Matching the record pairs

Within each block, every possible comparison pair \((d, m)\), \(d\) from the data file \(D\) and \(m\) from the master file \(M\), must be examined. In the examination of a pair, the normal procedure is to observe the extent of the agreement or disagreement separately for each of the matching variables. For any particular variable, the comparison outcomes can be coded or classified in a variety of ways. Suppose, for example, that one matching variable is sex, and that every record in both files has one of two codes for sex:

1. male
2. female

The outcome of a match could be restricted to:

1. agree on sex
2. disagree on sex

However, it may be desirable to extend this to use three categories:

1. agree on sex, sex is male
2. agree on sex, sex is female
3. disagree on sex

For some matching variables, such as surname, much more complex matching rules may be used, and there are many more possibilities. Although the phonetic compression code is normally used to block the file, (i.e. all the names that fall in Block S530 (SMITH etc)), the actual comparisons are based on the full surname. In some systems the comparison of the two names are based on a fixed number of characters, e.g. the first four characters of the surname (see Winkler, 1985b; Howe and Lindsay, 1981).

Still other systems use character-string comparison routines (string comparators, see Section 7.5) that take into account the likelihood of phonetic errors, transpositions of characters and random insertion, replacement and deletion of characters (Jaro, 1985, 1995; Winkler, 1993; Gill, 1997). Coding schemes for matching variables must also take into account the possibility that the variable may be missing for one or both members of the comparison pair, or that one or both are set to some not known or not collected value.

Generally, if resources permit, all the variables that are judged suitable for matching should be used in the computer comparisons. Where this is not possible, the variables can still be
employed later in manual verification where the outcome might otherwise be indeterminate. However, clerical intervention needs to be carefully limited and closely controlled. Clerical matching is extremely costly and, while individual clerical decisions can sometimes be better than those made by computer matching, humans usually lack consistency of judgement and can be distracted by extraneous information.

In most applications of the Fellegi-Sunter model the assumption is that the agreement (or disagreement) on one matching variable is independent of that on any other, conditional only on whether or not the records brought together are for the same person. To make this assumption plausible, special care needs to be taken in computing outcome specific weights for variables like sex and first name that are inherently related. Fellegi (1985) and Kelley (1986) have undertaken simulation studies to investigate the robustness of the U.S. Census Bureau's linkage system to violations of the independence assumption. For a particular population and a given set of linkage variables, they found that violations of the assumptions can have significant effects on the levels of matching errors. The elements of the combination of geographic variables like address, postcode and general practitioner are also inherently related.

7.10 Setting the matching threshold

7.10.1 Preliminary decisions on the linkages – the formal Fellegi-Sunter theory

To recapitulate, the heart of the matching system is the procedure that assigns each comparison pair to one of two or more linkage categories, based on the weights assigned in the previous stage. In the Fellegi-Sunter model that underlies our presentation in Section 7, the outcome weight is a function of the \( \log_2 \) likelihood ratio. In a probability-based system, the decision for each pair depends on the sum of the outcome weights for each matching variable. In the Fellegi-Sunter model the match weight is a function of the likelihood or probability ratio:

\[
\text{Likelihood ratio} = \frac{\text{Probability ratio}}{\text{Probability (result of comparison | given match)}} = \frac{\text{Probability (result of comparison | given non-match)}}
\]

The numerator represents the probability that the comparison of two records for the same person would produce the observed result. The denominator represents the probability that comparison of records for two different persons, selected at random, would produce the observed result. In general, the larger the ratio, the greater our confidence that the two records belong to the same person (see Section 7.3).

The graph shown as Figure 7.1 represents the distributions of the binit weights (logarithms of the overall likelihood ratios) for linked and the unlinked pairs in a particular study. The peak on the right of the graph represents the linked pairs and the peak on the left the unlinked pairs. The left peak is thousands of times larger than the right peak and for simplicity it has been truncated. The two distributions overlap between the points designated as the lower preset threshold and the upper preset threshold.

In the Fellegi-Sunter model, the comparison pairs are ordered according to the values of their weights. Two cut-off points are established. The higher of these separates the positive links from the possible links and the lower one separates the possible links from the positive non-links. These are shown as the upper preset threshold and the lower preset threshold respectively in figure 7.1. The establishment of appropriate cut-off values is a critical part of any record matching and linkage process. When a new record matching exercise is undertaken, all the matches on a small sample of the file are examined clerically. The threshold values are then determined. In subsequent runs these threshold values are used for
the determination of matches and non-matches. This process is recursive and the threshold values are so refined. At each stage the matches around the threshold values are rigorously checked by the clerks before the new and amended thresholds are used for the next run.

The objective of using the Fellegi-Sunter model is to place upper limits on the proportions of matched and unmatched pairs for which incorrect decisions are made. In choosing the target values $\mu$ (the wrongly matched or type II error) and $\gamma$ (the unmatched or type I error) (Annex A) for these proportions, one should be aware that the number of unmatched pairs is very much larger (thousands of times) than the number of matched pairs. Therefore, it is usually desirable to make $\mu$ considerably smaller than $\gamma$; otherwise the false matches will tend to swamp the false non-matches. In Figure 7.1, the shaded area from 0-10, denoted as missed matches, is a type I error $\gamma$ and the shaded area from 10 upwards, denoted as false positives, is a type II error $\mu$. Of course, the relative costs associated with resolving the two kinds of errors may also influence the choice of $\mu$ and $\gamma$.

As we have seen, some applications of model-based record-linkage systems require certain assumptions, such as independence of errors in the matching variables. Nevertheless, there is reason to believe that the Fellegi-Sunter procedure is in general fairly robust to departures from independence (but see Fellegi, 1985; Kelley, 1986). Moderate errors in the estimation of weights can lead to different linkage decisions only for comparison outcomes whose weights are close to one of the cut-off points.
Figure 7.1: Frequency distribution of bin weight for pairs of records.
Furthermore, there is no theoretical obstacle to extending the underlying models to take into account known dependencies between the linking variables (Kirkendall, 1985). There are also significant computational problems. Nevertheless, the approach is entirely workable, especially since the development of advanced linkage software which includes the work of Jaro and his collaborators (Jaro, 1985; Winkler, 1993; Winkler 1989a; 1993b; 2000; Larsen, 1996; Armstrong and Mayda, 1993; Armstrong 2000) (see also GRLS (Hill, 1990), the work of the ORLS (Gill et al., 1993, 1997) and Scotland (Kendrick, 1993).

7.10.2 Threshold setting and Type I and Type II errors

As previously described, the procedure for deciding whether two records belong to the same person is based on the total outcome specific weight, derived by algebraically summing the individual weights each of which is calculated from the comparisons of each pair of identifying variables on the data file and corresponding variables on the master file. This algebraic sum represents a measure of the (logarithm of the) likelihood that the two records are linked relative to the (log) likelihood that they are not linked. By comparing the total weight against a set of values which have been determined empirically, it is possible to decide whether the two records being compared actually refer to the same person.

Two types of errors can occur in record matching. The first type of error, false negative or missed matches or Type I error, is the more common and is a failure to match records which refer to the same person, possibly due to erroneous or missing variables on one or both records. Matches may also be missed if the two records fall into different blocks, which may happen where a surname is misspelled and the phonetic compression algorithm assigns the records into two different blocks. Consequently, the records which should have been assigned the same person number are instead assigned two or more person numbers and the records are not matched together.

The second type of error, false positive or wrongly matched records or Type II error, is less common but potentially more serious in assigning the same person number to two or more different persons. This type of error arises when the two records belonging to two different people have identifying sets that are almost identical. This situation arises where the two people have very common surnames and forenames, or are similar sexed twins. Problems also arise where a person has a number of forenames and they choose to use different forenames on different occasions. Unless other information is available with which to refute the match, the records are best left unmatched.

The frequency of both types of error is a useful measure of the reliability of the record matching procedure. These are of course related to the target values, \( \mu \) and \( \gamma \), of the Fellegi-Sunter theory that we have already discussed (see previous discussion about recommended relative values of these two error rates).

In preparing earlier versions of the ORLS linked files, a range of outcome weights was chosen and used to select records for clerical scrutiny. This range was confined by the upper and lower pre-set thresholds (see Figure 7.1). The false positive and false negatives are very sensitive to the threshold cut-off weight; too low, gives a very low false positive rate and a high false negative rate, too high, gives a high and unacceptable false positive rate with a low false negative rate. The values selected for the threshold cut-off is, of course, arbitrary, but must be chosen with care having considered the following objectives:

1. The minimisation of false positives, at the risk of increased missed matches.
2. The minimisation of missed matches, at the risk of increased false positives.
3. The minimisation of the sum of false positives and missed matches.
Threshold problems when combining name and non-name weights

The simple approach of algebraically summing the outcome weights, ignores the fact that the weight calculated for names is based on the degree of commonness of the name, and is passed on from other members of the family, whereas the weight for the non-names variables are based on distributions of those variables in the population, all values of which are equally probable. An unusual set of rare names information would generate high weights which would completely swamp any weights calculated for the non-names variables in the algebraic total, and conversely, a common name would be swamped by a perfect and identical set of non-names identifiers. This would make it difficult for the computer algorithm to differentiate between similarly named members of the population without resort to clerical assistance.

In the determination of the match threshold, a number of approaches have been developed, the earliest being the two stage primary and secondary match used in building the early ORLS files, through a graphical approach developed in Canada for the date of birth, to the smoothed two dimensional array approach developed by the ORLS and used for all its more recent matching and linking (Gill et al; 1987, 1993, 1997). Other advances in the methodology include the use of the EM Algorithm (Belin and Rubin, 1995; Larsen, 1996; Larsen and Rubin, 2000; Winkler, 1988, 1995; McLachlan and Krishnan, 1997) for the parameter estimation to determine the match thresholds.

Orthogonal mapping techniques in the ORLS system

Over the past ten years, ORLS (Gill et al, 1993, 1997) have developed an approach in which a two dimensional orthogonal array is prepared, analogous to a spreadsheet, with the algebraic sum of the names weights forming the abscissa ('X' axis) and the algebraic sum of non-names weights forming the ordinate ('Y' axis). In the development of the method, computer runs on sample data were undertaken and the pairs of records rigorously checked by very experienced clerks to determine whether the pairs did or did not match. The results of all these matches are stored in the cells of the orthogonal array designated by the co-ordinates (summed names weight, summed non-names weight). The empirical probabilities entered into the array were further interpolated and smoothed across the axes using linear regression methods and other curve fitting approaches including the use of cubic splines (Kelly, 1967; Hays, 1974; Borse, 1997).

Over 400,000 matches were rigorously checked by very experienced clerks and three counts were stored in each cell of the orthogonal array, designated by sum of names weight, sum of non-names weight:

(i) the total number of matches for that cell,
(ii) the number of good matches and
(iii) the number of non-matches.

A sample portion of this matrix is presented in Figure 7.2. One of the benefits of using two different axes for the matching threshold as described above is that a pair of records that contain a rare set of names or a perfect set of non-names information cannot be matched together unless there is good agreement on both axes. When a pair of records are being matched together, the matching software accesses the array and extracts the probability weight from the cell designated by the co-ordinates, as described above. The array of probabilities can be amended after each successive run, although minor adjustments or tinkering is discouraged. Precise scores and probabilities may vary, at least a little, according to the population and types of record pairs studied. A number of arrays have been prepared by the ORLS for the different types of event pairs being matched, for example, hospital to hospital records, hospital to death records, birth to hospital records, hospital and District Health Authority (DHA) records, cancer registry and hospital records, and so on.
A graphical representation of the array is shown in Figure 7.3, where each cell contains the empirical decision about the likelihood of a match between the record pair. The good matches are designated as ‘Y’, the non-matches as ‘N’ and the doubtful matches that require clerical intervention as ‘Q’. This graph is the positive quadrant of four quadrants where both the names and non-names weights are positive and greater than zero. In the microcomputer implementation of the software, this graph is held as a text file and can be edited using word-processing software.

Record pairs with weights that fall in the upper right part of the matrix are shown in Figure 7.3 as 'Y' are considered to be 'good' matches and a 1 per cent random sample of record pairs that fall near the Q-Y boundary are printed out for clerical scrutiny to measure the quality of the match. Record pairs with weights that fall between the upper and lower thresholds and shown in the figure as 'Q', are considered to be 'query' matches and all the record pairs are printed out for clerical scrutiny and the results keyed back into the computing system. Record pairs with weights falling below the lower threshold and shown on the map as 'N' are considered to belong to two different people, and a 1 per cent random sample of record pairs that fall adjacent to the N-Q boundary is printed out for clerical scrutiny.

At the end of each computer run, the results of the clerical scrutiny are pooled with all the existing matching results and new matrices can then be prepared. The strategy is to reduce the 'Q' zone to the minimum consistent with the constraints of minimum false positives (Type II errors) and false negatives (Type I errors). This will reduce the number of matches which require clerical intervention which is invariably the most costly and rate determining stage.

Further matrices have also been prepared that record the number of match variables used when matching a record pair, for example: (i) the number of surnames present, (ii) the number of forenames or initials and (iii) the numbers of other matching variables. Since the number of matrices can become quite large, intelligent systems and neural net techniques are being developed by ORLS for the interpretation of the N dimensional matrices and the determination of the match threshold (Ripley, 1997; Kasabov, 1996; Bishop, 1995: Baldi and Brunak, 1999).

Special procedures are required for the correct matching of similarly named same sex twins. Where the match weights fall within the clerical scrutiny area and the records are printed out,
Figure 7.2 Sample portion of the threshold acceptance array showing the number of matches and non-matches by outcome weight for names (X axis) and non-names variables (Y axis)

<table>
<thead>
<tr>
<th>WT=16</th>
<th>Percentage</th>
<th>Matches</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Nonmatches</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>37</td>
<td>41</td>
<td>45</td>
<td>58</td>
<td>83</td>
<td>77</td>
<td>91</td>
<td>98</td>
</tr>
<tr>
<td></td>
<td></td>
<td>198</td>
<td>177</td>
<td>231</td>
<td>255</td>
<td>319</td>
<td>277</td>
<td>413</td>
<td>298</td>
</tr>
<tr>
<td></td>
<td></td>
<td>537</td>
<td>255</td>
<td>298</td>
<td>145</td>
<td>65</td>
<td>83</td>
<td>41</td>
<td>4</td>
</tr>
<tr>
<td>WT=15</td>
<td>Percentage</td>
<td>Matches</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Nonmatches</td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
<td>---------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>41</td>
<td>38</td>
<td>42</td>
<td>56</td>
<td>61</td>
<td>75</td>
<td>87</td>
<td>98</td>
</tr>
<tr>
<td></td>
<td></td>
<td>190</td>
<td>223</td>
<td>211</td>
<td>316</td>
<td>410</td>
<td>329</td>
<td>218</td>
<td>522</td>
</tr>
<tr>
<td></td>
<td></td>
<td>273</td>
<td>364</td>
<td>293</td>
<td>245</td>
<td>265</td>
<td>109</td>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>WT=14</td>
<td>Percentage</td>
<td>Matches</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Nonmatches</td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
<td>---------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>18</td>
<td>25</td>
<td>21</td>
<td>19</td>
<td>31</td>
<td>56</td>
<td>77</td>
<td>93</td>
</tr>
<tr>
<td></td>
<td></td>
<td>113</td>
<td>87</td>
<td>90</td>
<td>110</td>
<td>190</td>
<td>198</td>
<td>660</td>
<td>422</td>
</tr>
<tr>
<td></td>
<td></td>
<td>514</td>
<td>261</td>
<td>330</td>
<td>460</td>
<td>412</td>
<td>162</td>
<td>197</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WT=10</td>
<td>Percentage</td>
<td>Matches</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Nonmatches</td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
<td>---------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>7</td>
<td>8</td>
<td>14</td>
<td>14</td>
<td>11</td>
<td>22</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td></td>
<td>17</td>
<td>35</td>
<td>28</td>
<td>34</td>
<td>50</td>
<td>50</td>
<td>69</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td></td>
<td>341</td>
<td>404</td>
<td>284</td>
<td>382</td>
<td>277</td>
<td>295</td>
<td>235</td>
<td>203</td>
</tr>
<tr>
<td>WT=9</td>
<td>Percentage</td>
<td>Matches</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Nonmatches</td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
<td>---------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>8</td>
<td>12</td>
<td>13</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>18</td>
<td>42</td>
<td>28</td>
<td>47</td>
<td>64</td>
<td>90</td>
<td>90</td>
<td>87</td>
</tr>
<tr>
<td></td>
<td></td>
<td>757</td>
<td>966</td>
<td>637</td>
<td>952</td>
<td>706</td>
<td>644</td>
<td>588</td>
<td>474</td>
</tr>
<tr>
<td>WT=8</td>
<td>Percentage</td>
<td>Matches</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Nonmatches</td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
<td>---------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>7</td>
<td>7</td>
<td>9</td>
<td>12</td>
<td>16</td>
<td>20</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td></td>
<td>95</td>
<td>70</td>
<td>70</td>
<td>118</td>
<td>113</td>
<td>140</td>
<td>140</td>
<td>170</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1,234</td>
<td>812</td>
<td>812</td>
<td>1,106</td>
<td>785</td>
<td>728</td>
<td>583</td>
<td>588</td>
</tr>
<tr>
<td>WT=7</td>
<td>Percentage</td>
<td>Matches</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Nonmatches</td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
<td>---------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
<td>Ret</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>45</td>
<td>43</td>
<td>55</td>
<td>58</td>
<td>57</td>
<td>68</td>
<td>93</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2,721</td>
<td>3,919</td>
<td>2,723</td>
<td>3,576</td>
<td>2,458</td>
<td>2,542</td>
<td>1,952</td>
<td>1,848</td>
</tr>
</tbody>
</table>

Source: Gill, 1997

the clerks should identify the two records involved, mark the records in some agreed fashion and amend the match where necessary. These procedures are also required for elderly people who are recorded in the information system under a given set of forenames but, on a subsequent hospital admission or when they die, a different set of forenames are reported by the person themselves, or by the next of kin.
Use of orthogonal matrices to cope with different numbers of matching variables

To some extent the use of orthogonal matrices based on a particular record type can take into account the vagaries of that dataset, however some of the variables may not be entered into the matching system. For example, some records may contain birth surname and for other types of record or period of time, this variable may not have been collected. The summed outcome weight would contain the birth surname contribution for some matches and not for others and so some of the records may fail to be matched together.

Amended orthogonal arrays can be prepared for the number of variables that are entered into the matching process. Birth surname and second forename or initials are variables that are
most likely to be missed by some data capture systems, but are the most valuable for accurate matching.

7.11 Resolution of uncertainties: how to deal with the queries and errors in matching

After the preliminary linkage decisions have been completed, there are usually some uncertainties that need to be resolved. These consist of pairs that have been classified as possible links or of multiple links, i.e. groups of linked pairs that have one or more records in common.

In the Fellegi-Sunter procedure, possible links are the pairs that fall between the upper and lower cut-offs, in the ORLS model these records are assigned to the 'Q' matrix weight. If resources permit, these pairs may be reclassified as positive links or non-links either by collecting more data or by clerical review of the record content for these pairs.

If statistical estimates are to be made, and the resources needed to seek further information are not available, the potential links may be treated as non-links and a survey-type non-response adjustment may be made (Scheuren, 1980). It is also possible to consider keeping some of the potential links and then conducting the analysis, with an adjustment being made for mismatching (Scheuren and Oh, 1975).

Multiple links can occur in the Fellegi-Sunter formulation because the linkage decision is made independently for each pair. As a result, a record from either file may be included in more than one pair whose weight exceeds the cut-off for a positive link. In some applications, these many-to-one links might be appropriate, but usually a further step has to be taken to select the 'best' one using a linear assignment algorithm.

Clerical procedures

The clerical review is usually the best basis for the final matching decisions, particularly where further information is sought or is available to help in this process. Some automated systems provide preliminary indications of the record pairs judged to be the likely candidates. The National Death Index (NDI) operating system leaves it to users to resolve indeterminate cases. For each user record, they list as possible links all death records that qualify under one or more of 12 sets of matching criteria (e.g. agreement on SSN and first name, agreement on SSN and last name, agreement on month and day of birth and first and last names, etc). Users with small files usually resolve multiple links by clerical review. For large studies, some users have developed their own computer algorithms for this purpose (Patterson and Bilgrad, 1985). Users must also be prepared to determine final match status when only one possible link has been identified for a name submitted to the NDI. Doing this may often be more difficult than resolving multiple links. Jaro (1985) offers a computerised transportation algorithm to solve multiple linkage problems. His approach is most effective when all the linking information has already been computerised and when there are contentious problems in the linkages, that is, 'n' records on one file are matching 'm' records on another. The procedure is analogous to the 'constrained matching' approach used in statistical matching, i.e. it picks a single best set of matches, rather than picking the best match for each record in one of the input files. Armstrong (2000) describes an alternate 1-1 matching procedure used at Statistics Canada. It is essentially a greedy heuristic. Winkler (1994) gives details of a generalised assignment procedure that works better than the Burkard and Derigs assignment procedure.

ORLS undertake a clerical review of all the matches that fall in the review area (see Section 7.11). The doubtful matches are printed out and the clerks compare the variables that were used for this match. In some cases the computer decision can be reversed especially where the
clerk can resolve the partial information. In cases where there is some doubt, the records are always left unmatched.

Matching is only the first stage in the record linkage process, the second stage is linking the matched record with the other records for this person or entity. Rules can be prepared using Boolean constructs, for example, the gender on both records must be the same, or the date of birth must be within one year. If the rules are satisfied the matched record can be linked in with the existing records, where there is some conflict, the whole set of records will need to be printed out for clerical scrutiny. This is especially the case where there is some error in the dates since the record may be inserted in wrong temporal sequence.

Even when many-to-one links are not appropriate, in theory, it may be desirable to use the additional information they provide, especially if conditions do not permit a clear determination of which of the links represent true matches. Suppose, for example, that a record in File A is initially classified as a positive link with each of three records in File B. Three linked records could be established, each associating the File A record with one of the positive links from File B. The outcome weight associated with the File A record would be divided among the three linked records: we might allocate one-third of it to each linked record or we might prefer to allocate it in proportion to the weights used in making the initial linkage decisions.

Difficulties with indeterminate cases can often be traced back to design flaws in the data linkage system. For example, not enough linking information may have been obtained on one or both files to assure uniqueness. The degree of redundancy in the identifiers may have been insufficient to compensate completely for the reporting errors. In an operational context, the record matching and linkage process may be so constrained by costs that, even if there are sufficient linkage variables, they cannot be adequately exploited.

7.12 Consequences of errors in matching, Type I and Type II errors

In most record-linkage studies, the matching is performed in a conservative manner with regard to the links that should be accepted. Sometimes this may mean additional expense to obtain more information, or the risk of seriously biasing results by leaving out a large number of the potential links. In any event, further research is needed on applying more complex analytical techniques that take explicit account of the false match rate, possibly by examining the errors in variable where the false match rate is estimated, e.g. as in Scheuren and Oh (1975). This would allow a correction factor to be derived. Attempts to find methods of estimating the false match rate should also be undertaken (Belin and Rubin, 1995; Winkler, 1988; McLachlan and Krishnan, 1997).

7.13 Combining results from many match runs using different blocking keys

File blocking enables all records having the same value in the blocking variable to be compared. One consequence of this strategy is that records not having this particular value in the blocking variable will automatically be classified as a non-match. In fact, if our blocking variable was age, and the age on one of the records was in error, then it would be considered as unmatched. To get around this problem, multiple passes are used, which are based on different blocking variables.

The blocking strategies for each pass should be as independent as is possible. The data file and the master file should be blocked and sorted on any of the matching variables, for example: (i) present surname order, (ii) birth surname order, (iii) date of birth order, (iv) forename order, or, (v) postcode order and so on. The results from each of these quite
independent matches can then be combined together and duplicate entries removed. The output from these matches will consist of a file of records, each of which holds two person numbers, one from the data file record and one from the master file record. This file can then be used to amend or update the linking of the data file to the master file, or to update the index.

It is recommended that every record on the data file and the master file should be allocated a unique person or entity number. When the data record matches with a record on the master file a link is created between the two records, each of which would normally have a different person number. In some systems the two person numbers will be recorded in an index, while in other systems, the person number from one record will be copied over the person number of the second record, both records then sharing the same person number.

Where there is a one-to-one correspondence between the person numbers on the two records, the match can be readily accepted. Where there is a one-to-many or a many-to-one arrangement either the best matches (the one with the highest outcome weights) can be accepted or all the matches should be printed out for clerical scrutiny. Records which cannot be sequenced into the same Soundex block (for example HORTON (H635) and HAWTON (H350)) can be blocked under other identifiers for example: date of birth, postcode or forename and matched in the normal way, and the results combined with matches from other blocking orders.

7.14 Reducing risks associated with wrong matching

The major risks in using data that is badly matched is that either the data records will not be matched or will be linked with the records for a different person. In most cases this bad match can be detected when a logical check is performed across all the records for any given person. Sometimes it is almost impossible to determine whether the record belongs to the person or not. In these cases, the link cannot be made and the data record is then regarded as belonging to a new person. The risks can be reduced by taking great care in the following parts of the matching process:

1) Selection of the matching variables and the order in which they will be used in the matching process. The general rule is that the variables that are universally available should be used for blocking, bearing in mind that the variables should also be fixed, accurately recorded and have a high discriminating power.

2) Check that the phonetic coding algorithm can cope with the different surnames in the population, since many spelling variations for common surnames may fall into separate blocks. The variations for a given surname e.g. HAWTON and HORTON can then be stored in an agreed block using a lexicon (see Section 5.2).

3) Use a lexicon for the conversion of forenames and other matching variables to a standardised format.

4) Calculate the outcome specific weights for the population that you are matching, since there are wide variations in the frequency of common surnames and forenames across a geographical area.

5) Carry out many test runs with different values of the matching threshold criteria using a random sample of the data records. This will involve clerical checking of the matches made, but the time spent in this activity will be offset by the better matching and the lower future clerical requirement.

87
7.15 Typical match rates

During the course of a number of studies on hospital inpatient records and mortality, it was found that the number of links lost through problems in the method of blocking the files using names identifiers, ranged from two per cent for most studies through to 30 per cent where the identifiers were poor. These estimates were quantified by re-matching the file on a number of different keys, and also examining the file for certain events that are expected to happen together. Examples of these are: looking for a matched death certificate for a person who has died in hospital, and looking for the second record where a person has been transferred from the main hospital to another hospital. Less likely are the false positive matches arising from the correspondence of the identifiers from two different people.

<table>
<thead>
<tr>
<th>Box 7.14 Typical results obtained from probabilistic matching of datasets containing names or other matching variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Matching the ORLS dataset (all record types) Using ONCA/Year of Birth (Gill 1987)</td>
</tr>
<tr>
<td>2. Matching the ORLS dataset (hospital records vs hospital records) Using ONCA/Year of Birth (Gill 1987)</td>
</tr>
<tr>
<td>3. Matching the ORLS dataset (hospital records vs death records) Using ONCA/Year of Birth (Gill 1987)</td>
</tr>
<tr>
<td>4. De-duplication of NHSCR Probability match using Surname/forename/sex/date of birth (Gill, 1994)</td>
</tr>
<tr>
<td>5. De-duplication of NHSCR Probability matching using the ONCA code (Gill, 1997)</td>
</tr>
<tr>
<td>6. Probability matching of an ORLS (HES) dataset against itself using the ONCA code (Gill, Goldacre,McGuiness, 2000)</td>
</tr>
</tbody>
</table>

Source: Various
8. Methods of building the linked file from the matched records

8.1 Building the linked files

It is a common practice in record linkage to match the data file against the master using different blocking criteria. The output from each of these matching runs is a text file that contains details about each pair of matched records. The separate output files of matched records should be combined and merged into one output file and the entries used to update the index or amend the data file. The single output file then can be sorted so that all the records for the same person appear consecutively.

There are two main ways for of building the single linked file, in the first method a single unique person number is applied to all the matched records and in the second method links are set using the accession number in an index. These records can then be back loaded into the database and all records for the same person or entity can be sorted or indexed and analysed together.

In the ORLS system, the data file is matched against the master file using many different blocking orders, for example:

(i) present surname,
(ii) phonetic compression of the present surname,
(iii) birth surname,
(iv) first forename or initial,
(v) date of birth,
(vi) NHS number and other numbers, e.g. Hospital + Hospital Unit number
(vii) Year of birth/gender/postcode
(viii) Date of delivery and date of birth.

The output files generated by the matching system contains the variables listed in Box 8.1. The number of records written to the output file for any one person can be very large, and is approximately the number of records that matched on the data file multiplied by the number of matched records on the master file. Combinatorial and heuristic algebraic methods have been developed for the reduction of all similar records in a set down to a small number, ideally one for each match pair (Hu, 1982; Cameron, 1994; Lothaire, 1997; Reeves, 1995)

The building of the linked files in Oxford Record Linkage Study (ORLS) is accomplished by allocating the same person number to all the matched records for a given person and this rule is crucial to the whole process of person or entity matching. The details of the procedure are explained in the following.

1) Let A1, A2, A3 ... be records on the master file matched to the record A on the query file. If the person corresponding to the record A, A1, A2, A3... has not changed his name(s), sex, date of birth and possibly address, the person number for all the records A, A1, A2,A3...is set to a common value which is the lowest of the person numbers for A1, A2 or A3. This is illustrated with an example in Box 8.2. The four person records shown in the example in Box 8.2 will start off with four different person numbers (designated as 1451,1796,1845,8735). After the matching and linking stages have been completed, all four records will have links set to the lowest person number, either through an index, or by overwriting the person number on records 2,3,4 with 1451. The accession number could also be used in an index to indicate that records with accession numbers 1649, 2136, 3798, all link to the record with accession number 1237. In most cases it is better to use the person number
than the accession number, since collating all the records for a given person or entity is just a sorting process.

**Box 8.1 Part outputs from the matching runs in the ORLS system**

<table>
<thead>
<tr>
<th>Details of Data record</th>
<th>Person number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accession number</td>
</tr>
<tr>
<td></td>
<td>Record type</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Details of Master file record</th>
<th>Person number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accession number</td>
</tr>
<tr>
<td></td>
<td>Record type</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Details about the match run</th>
<th>Output print stream (good match or query match)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sum of the names weights</td>
</tr>
<tr>
<td></td>
<td>Sum of the non-names weights</td>
</tr>
<tr>
<td></td>
<td>Number of variables used in this match</td>
</tr>
<tr>
<td></td>
<td>Cross-reference to the clerical print-out</td>
</tr>
<tr>
<td></td>
<td>Clerical matching decision (either Y or N)</td>
</tr>
</tbody>
</table>

2) Where a person has a change of name, for example a single woman gets married within the span of the file, there may be two types of records on the query file and the master file, one corresponding to her details before the change and the other corresponding to after the change. Correspondingly the matched records for the person will be grouped into, those recorded under her birth surname (maiden name) say A, A1, A2, A3,..., and those recorded under her married name (present surname), say C, C1, C2, C3,... The common person number allocated to A, A1, A2, A3 will be the lowest of the person numbers for As and the common person number allocated to C, C1, C2, C3 will be the lowest of the person number for Cs. This sequence of events, and examples of typical person records are shown in Box 8.3. Person records 1–4 will be linked under person number 1451 and person records 5–8 will be linked under person number 5451.

Although matching using date of birth may indicate that A and C refer to the same person, if there is any doubt, the two sets As and Cs should not be linked together at this time. Subsequently matched records will link to either her single or married records, one of which will carry the person number for the group A, A1, A2,A3, and the other for the group C, C1, C2, C3,... It is also possible that there are records on the master file say B1, B2 which contain details of the person before and after the change of name. In this case the query record A will also be linked to B1, B2 and likewise C will be linked to B1, B2 and hence all the records As, Bs and Cs should be linked. The new unique person number for all the records As, Bs and Cs will be set to the lowest of the individual person numbers and this is illustrated in Box 8.4.
Box 8.2 Sequence of records with same Person Number

Sequence of the records for a person:

A1, A2, A3 etc
A1 = A2 = A3 = A (= signifies match)

Examples of typical person records:

<table>
<thead>
<tr>
<th>Birth Surname</th>
<th>Present Surname</th>
<th>First Forename</th>
<th>Second Forename</th>
<th>Sex</th>
<th>Date of Birth</th>
<th>Address</th>
<th>Accession Number</th>
<th>Old Person Number</th>
<th>Final Person number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>Hall</td>
<td>Margaret</td>
<td>Elizabeth</td>
<td>F</td>
<td>220749</td>
<td>14,High</td>
<td>1237</td>
<td>1451</td>
<td>1451</td>
</tr>
<tr>
<td>Smith</td>
<td>Hall</td>
<td>Margaret</td>
<td>Elizabeth</td>
<td>F</td>
<td>220749</td>
<td>14,High</td>
<td>1649</td>
<td>1796</td>
<td>1451</td>
</tr>
<tr>
<td>Smith</td>
<td>Hall</td>
<td>Margaret</td>
<td>Elizabeth</td>
<td>F</td>
<td>220749</td>
<td>14,High</td>
<td>2136</td>
<td>1845</td>
<td>1451</td>
</tr>
<tr>
<td>Smith</td>
<td>Hall</td>
<td>Margaret</td>
<td>Elizabeth</td>
<td>F</td>
<td>220749</td>
<td>14,High</td>
<td>3798</td>
<td>8735</td>
<td>1451</td>
</tr>
</tbody>
</table>

Alternatively, it is also possible that A will get matched to C1, C2, ... and C will get matched to A1, A2, ... when different blocking criteria are used. In this case all the records As and Cs will be assigned the unique person number as illustrated above.

In the examples shown in Box 8.4, recursive matching the records will generate \(10 \times 9 / 2 = 45\) runs the results of each will be pairs of person numbers i.e.,

- pair 5,4 5735 8735
- pair 1,4 1237 8735
- pair 8,1 6796 1451 etc.

these pairs can then be reduced using combinatoric and heuristic techniques to the lowest value of the person number, which in this example is 1451 (Hu, 1982; Cameron, 1994; Lothaire, 1997; Reeves, 1995).

3) Where the person has had many changes of name or marital status, the number of different types of links will increase. Over the 38 year span of the ORLS system, links up to 5 deep have been found. Unless bridging records are recorded in the file systems, i.e., birth surname/present surname, present surname1/present surname2 etc., the person would have their records divided into five parts. In matching exercises on files containing names from some ethnic communities, links up to 10 deep have been found. These have occurred since it is difficult to determine the naming practices used by the various communities.
Box 8.3 Sequence of records for a person who has two Person Numbers

Sequence of the records for this person: A1, A 2, C1, C2, A3, C3 etc.

These will be linked as:

\[
\begin{align*}
A1 & = A2 = A3 = A \\
C1 & = C2 = C3 = C
\end{align*}
\]

As being links under her birth name

Cs being links under her married name

Examples of typical person records:

<table>
<thead>
<tr>
<th>Birth Surname</th>
<th>Present Surname</th>
<th>First Forename</th>
<th>Second Forename</th>
<th>Sex</th>
<th>Date of Birth</th>
<th>Address</th>
<th>Accession Number</th>
<th>Old Person number</th>
<th>Final Person number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Hall</td>
<td>Margaret</td>
<td>Elizabeth</td>
<td>F</td>
<td>220749</td>
<td>14, High</td>
<td>1237</td>
<td>1451</td>
<td>1451</td>
<td></td>
</tr>
<tr>
<td>2 Hall</td>
<td>Margaret</td>
<td>Elizabeth</td>
<td>F</td>
<td>220749</td>
<td>14, High</td>
<td>1649</td>
<td>1796</td>
<td>1451</td>
<td></td>
</tr>
<tr>
<td>3 Hall</td>
<td>Margaret</td>
<td>Elizabeth</td>
<td>F</td>
<td>220749</td>
<td>14, High</td>
<td>2136</td>
<td>1845</td>
<td>1451</td>
<td></td>
</tr>
<tr>
<td>4 Hall</td>
<td>Margaret</td>
<td>Elizabeth</td>
<td>F</td>
<td>220749</td>
<td>14, High</td>
<td>3798</td>
<td>8735</td>
<td>1451</td>
<td></td>
</tr>
<tr>
<td>5 Smith</td>
<td>Margaret</td>
<td>Elizabeth</td>
<td>F</td>
<td>220749</td>
<td>14, High</td>
<td>1256</td>
<td>5451</td>
<td>5451</td>
<td></td>
</tr>
<tr>
<td>6 Smith</td>
<td>Margaret</td>
<td>Elizabeth</td>
<td>F</td>
<td>220749</td>
<td>14, High</td>
<td>1692</td>
<td>6796</td>
<td>5451</td>
<td></td>
</tr>
<tr>
<td>7 Smith</td>
<td>Margaret</td>
<td>Elizabeth</td>
<td>F</td>
<td>220749</td>
<td>14, High</td>
<td>2165</td>
<td>7845</td>
<td>5451</td>
<td></td>
</tr>
<tr>
<td>8 Smith</td>
<td>Margaret</td>
<td>Elizabeth</td>
<td>F</td>
<td>220749</td>
<td>14, High</td>
<td>3998</td>
<td>8741</td>
<td>5451</td>
<td></td>
</tr>
</tbody>
</table>

8.2 Sorting and logically checking the records

The resulting linked files in the ORLS system, are sorted on: (i) person number, (ii) record type and (iii) date of the event. The sequence of records for any person will start with their birth and end with their death, and all the intervening records should form a logical time ordered sequence. Various checks can then be performed to ensure that the dates for the person are logically correct, i.e. the date of birth is before the date of death and that the various types of life events are in a logical temporal sequence.

Errors in the system due to administrative procedures need to be treated with caution, since on some occasions a person who died on a Saturday will be administratively discharged dead from hospital on the Monday. Other factors that need to be examined include: a person being in two types of inpatient health care at the same time, or being admitted to a second hospital before being discharged from the first hospital. When these cases are found, the appropriate corrective action needs to be taken.
Box 8.4 Sequence of records for a person who might have three or more different Person Numbers

Sequence of the records for this person: A1, A2, B1, B2, C1, C2 etc.
These will be linked as: A1 = A2 = B1 = B2 = A, B1 = B2 = C1 = C2
A being linked under her birth surname
B being links under her birth surname and her married name
C being links under her married name

The person numbers on these records will be recursively converted to;
A1 = A2 = B1 = B2 = A, C1 = C2 = C
And then to,
A = A = A = A = A etc.
A being links under her birth surname
B being links under her birth surname and her married name
C being links under her married name

Examples of typical person records:

<table>
<thead>
<tr>
<th>Birth Surname</th>
<th>Present Surname</th>
<th>First Forename</th>
<th>Second Forename</th>
<th>Sex</th>
<th>Date of Birth</th>
<th>Address Number</th>
<th>Accession Old Person-number</th>
<th>Final Person-number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Hall</td>
<td>Margaret</td>
<td>Elizabeth</td>
<td>F</td>
<td>220749</td>
<td>14,High</td>
<td>1237</td>
<td>1451</td>
</tr>
<tr>
<td>2</td>
<td>Hall</td>
<td>Margaret</td>
<td>Elizabeth</td>
<td>F</td>
<td>220749</td>
<td>14,High</td>
<td>1649</td>
<td>1796</td>
</tr>
<tr>
<td>3</td>
<td>Hall</td>
<td>Margaret</td>
<td>Elizabeth</td>
<td>F</td>
<td>220749</td>
<td>14,High</td>
<td>2136</td>
<td>1845</td>
</tr>
<tr>
<td>4</td>
<td>Hall</td>
<td>Margaret</td>
<td>Elizabeth</td>
<td>F</td>
<td>220749</td>
<td>14,High</td>
<td>3798</td>
<td>8735</td>
</tr>
<tr>
<td>5</td>
<td>Smith</td>
<td>Hall</td>
<td>Margaret</td>
<td>F</td>
<td>220749</td>
<td>14,High</td>
<td>9623</td>
<td>5735</td>
</tr>
<tr>
<td>6</td>
<td>Smith</td>
<td>Hall</td>
<td>Margaret</td>
<td>F</td>
<td>220749</td>
<td>14,High</td>
<td>8542</td>
<td>3735</td>
</tr>
<tr>
<td>7</td>
<td>Smith</td>
<td>Margaret</td>
<td>Elizabeth</td>
<td>F</td>
<td>220749</td>
<td>14,High</td>
<td>1256</td>
<td>5451</td>
</tr>
<tr>
<td>8</td>
<td>Smith</td>
<td>Margaret</td>
<td>Elizabeth</td>
<td>F</td>
<td>220749</td>
<td>14,High</td>
<td>1692</td>
<td>6796</td>
</tr>
<tr>
<td>9</td>
<td>Smith</td>
<td>Margaret</td>
<td>Elizabeth</td>
<td>F</td>
<td>220749</td>
<td>14,High</td>
<td>2165</td>
<td>7845</td>
</tr>
<tr>
<td>10</td>
<td>Smith</td>
<td>Margaret</td>
<td>Elizabeth</td>
<td>F</td>
<td>220749</td>
<td>14,High</td>
<td>3998</td>
<td>8741</td>
</tr>
</tbody>
</table>

8.3 Other outputs from the matching process

Other outputs will include statistics for the matching linking process. Often a subset of the records is extracted from the master file that only contains the variables that will be used in the analyses, together with any derived variables. For example, to support further analyses the extracted records could include the following derived variables:

- a) Area classification or deprivation score derived from postcode;
- b) Data borrowed across of all the records for a given person, where there are gaps in the records;
- c) Smoothed data for some variables, for example date of birth, where there is some noise in the raw variables (e.g. heaping of dates);
- d) Aggregations of raw variables where this would help in the running of subsequent analyses, using for example when using SPSS® or SAS®.
9. **Commercial software packages available for record linkage**

There are a number of commercial, government and educational establishments that offer software for record matching and linking. The following list, while not exhaustive, is a good indicator of what is available in the market place, **based solely on supplier information**.

1) **SSA NAME3**

SSA Name 3 is designed to match name data which can either refer to people or to organisations. The software does not require any data cleaning before the matching is undertaken. The software also provides routines for the parsing and matching of addresses. The software works by formatting the data file by the removal of erroneous characters and standardising the matching variables, then using a phonetic approach to generate a software key. The keys are stored in a separate database and are based on a set of rules that apply to the given dataset. The system can create more than one keys to allow for variations in the matching variables. The software will return the references between the keys and the original data file. The software ranks the data in order of the likelihood of a match. A new key needs to be created for each data file. The software comes as a collection of routines that need to be built into existing software by separate developers. The product runs on all the usual computing systems. The software is prepared and marketed by **Software Search America**, for further access the internet site [www.searchsoftware.com](http://www.searchsoftware.com)

2) **INTELLIGENT SEARCH TECHNOLOGY Ltd**

The software offered by Intelligent Search Technology is a name and address search and matching software. The product runs on all the usual computing systems. The applications include record retrieval, merge and purge files, detection and removal of duplicate records. The software is prepared and marketed by **Intelligent Search Technology Ltd**, for further details see the internet site [www.intelligentsearch.com](http://www.intelligentsearch.com)

3) **INTELLIGENT RETRIEVAL**

Intelligent retrieval software is designed to match many different types of dataset from names and addresses in the market sector. The product has been specifically designed for on-line use. The software creates a series of keys based on the descriptive data within the file and stores these in a database. Using intelligent methods which use name synonyms, phonetics and spelling comparisons, acronyms and abbreviations the software can effect a match across the data files. The software will return the links to the original source data, these must then be processed by an application program to retrieve and link the data. The software is prepared and marketed by **Human Inference**, for further details access the internet site [www.humaninference.com](http://www.humaninference.com)

4) **TRILLIUM**

Trillium is part of a suite of software tools and the tool of interest is the Matcher. This works on line and does not use any special keys. This approach supports a flexible definition of the matching keys together with a rule based matching algorithm. The software includes a parser for cleaning up the data and a geocoder for verifying the address information. The matching algorithm includes the use of phonetics, spelling and name synonyms and abbreviations and any combination of these factors can be used to effect the match. The outcome from the matcher include Pass, Query or Fail. The software is prepared and marketed by **Harte Hanks**, for further details access the internet site [www5.harte-hanks.com](http://www5.harte-hanks.com).
5) QUICK ADDRESS

The Quick-Address range offers a software solution designed for name and address management. The software includes routines for name and address capture, cleaning and enhancing the addresses, and the application of geographical information systems. The software is prepared and marketed by QAS Systems, for further details access the internet site www.qas.co.uk

6) INTEGRITY (includes the Matchware Technologies Automatch software)

The INTEGRITY Data Re-engineering product is aimed at improving corporate data and the incorporation of the data into warehouses. In 1998 Vality acquired the Matt Jaro Matchware Technologies Inc record matching tools. The Integrity suite offers a set of record matching tools that can be used for on-line searching. The software includes a variety of matching algorithms, which include the Soundex and NYSIIS phonetic searching and blocking software. The software can also cope with spelling variations and abbreviations in the text strings. The software is prepared and marketed by Vality Systems, for further details see the internet site www.vality.com

7) PA OYSTER ENGINE

The PA consulting group have developed a simple fuzzy matching engine that can be integrated into the systems that they have developed. The system uses an online strategy, can carry out several data cleaning and matching operations and can undertake probabilistic record matching. The software is prepared and marketed by PA Consulting Group, for further details see the internet site www.pa-consulting.com/it_service/index_stan.html

8) ALTA

The software offered by this company include data analysis for the detection and removal of duplicate medical records. The product runs on all the usual computing systems. The applications include record retrieval, merge and purge files, detection and removal of duplicate records. The ALTA probabilistic algorithms were first developed in 1988. Since then they have pioneered the use of such algorithms to detect multiple medical record numbers for the same person. Using the ALTA algorithms, pairs of records can be linked together that potentially represent the same person and assigned a confidence weight to each match. The software is prepared and marketed by Advanced Linkage Technologies of America Inc, a division of Madison Information Technology, for further details access the internet site www.altascan.com.

9) GRLS 3

This product addresses the problem of trying to link records where no unique identifiers exist. The software suite use the probabilistic record matching methodology and can be used for de-duplication of data files or matching together records from a number of data files. GRLS 3 uses a client server architecture with a UNIX server. The software is prepared and marketed by Statistics Canada, for further details email: ted-hill@statcan.ca.

10) OXLINK

This product was specifically designed to matched together the various data files for the creation of the Oxford Record Linkage Study (ORLS). The software suite includes routines for the parsing and blocking of the data files, the generation of the phonetic codes used for blocking, and uses both the exact and probabilistic record matching methodologies. The software suite can be used for de-duplication of data files or matching together records from a number of data files. The software can cope with encrypted files provided that the decryption
routines and the relevant keys are entered into the system at run time. The software is written in Fortran and C and processes flat clustered files in character format. The implementation can be run on both UNIX servers and the full range of desk top PCs. The software is prepared and marketed by the University of Oxford, for further details contact at the address email: leicester.gill@dphpc.ox.ac.uk.

11) Census Bureau (Bill Winkler)

This product is a set of record matching and linking routines written by the US Bureau of the Census for use in census matching. The software is written in Fortran and C and processes flat files in character format. The implementation can be run on both UNIX servers and the full range of desk top PCs. The software is prepared and marketed by the United States, Bureau of the Census, for further details contact at the address email: Winkler@census.us

12) SPSS, SAS, MS Office

All of these software suites are capable of undertaking a exact match using the variables in each record. Use of compound keys is very problematical

13) Some SAS routines for Probabilistic linkage are available from the internet site, http://www.samhsa.gov/csat/idbse/idblink.asp or email: info@samhsa.gov
10. Typical results from the selected GSS linkage projects

To ascertain the practical issues involved in record matching and linkage, and identify potential benefits of record linkage in official statistics it was decided to study in depth a sample of record linkage applications uncovered in an earlier survey ‘Record Matching Applications in GSS’. A sample of case studies were selected and a detailed questionnaire circulated to the named contacts for the case studies. The questionnaire requested information on: the types of datasets matched, the organisation undertaking the matching, matching environment and software employed, sequencing and pre-processing of data files, matching method used and quality of matching and costs. The details of the responses are given in the Appendix F. The salient features of these studies are noted below:

1) The Comprehensive business directory

It is an application currently under consideration, aiming to link data from a variety of sources such as PAYE data, VAT information and Companies House information in order to create 'Single Business Register'. The largest data set is expected to have 1.6 million records and would possibly use the probabilistic record matching methodology. The project feasibility study has been commissioned from the IBM/Vality consortium. The potential benefits include reducing respondent burden, improving data quality and standardisation.

2) One Number Census Matching

This is a planned application aiming to link 2001 Census data with the Census Coverage Survey (CCS) data. The largest data set is expected to be about 56 million records and the matching is expected to be a combination of exact and probabilistic matching methods. Currently, an in-house system based around Sybase and C++ is being developed. The benefits of matching include the ability to estimate the under-enumeration in the decennial census and thereby improving census data quality.

3) Retail planning and GIS

This is an on-going application that links together data from a number of sources including the Inter-Departmental Business Register (ONS), Floor Space data (Valuation Office Agency), Code Point (OS) and the Central Postcode Directory. The aim of matching is to create statistics of town centre activity. The matching uses exact methodology, and some proprietary software. The project was commissioned from University College London, which carries out matching on an annual basis. The benefits of matching include improving data quality for retail planning.

4) Secondary school performance tables and key stage 3 to GCSE value added

It is an on-going application and the purpose of the matching is to create a single data base of qualifications by linking together Key Stage 3 results, GCSE results and GNVQ results for each pupil. The method of matching employed is exact using the pupil number as the key. The project has commissioned the University of Bath to undertake the matching and they have developed proprietary software for this task. The benefits of matching include reducing burden on schools of the performance tables by providing them with a pre-matched set of results which they can check, and also improving data quality.

5) Cross matching of the NHSCR against itself

The objective of this study was to match the NHSCR against itself to estimate the number of duplicate entries on the register. The NHSCR register was built from the 108 files prepared by each of the FHSAs (now DHAs) and merged together in June 1990. Some of the people on
the NHSCR will have been included from two or more of the individual registers, under slightly different identifiers. People who have left the UK for short periods and subsequently returned are issued with a new NHS number. A complete sample of the NHSCR was extracted and prepared for internal cross matching. The parallel files containing the changes in names, date of birth and address were also extracted. The files were merged and expanded as described elsewhere in this report (Section 5.2), the 57 million records upon expansion and merging resulting in 81 million records. The dataset was internally matched using a combination of exact and probabilistic methods. The output being a file of those people who have two or more entries in the NHSCR.

6) Building a linked file of hospital records (ORLS system)

The objective of this study is to match together hospital discharge records and death registration details to prepare a file suitable for a number of analytical procedures. The matching was undertaken as part of the ORLS file building programme and the variables, which include names and addresses, are extracted from both the data and master files and used in the probabilistic match. The results of the match are applied to the data file and it is split into two parts, the name and address variables (and no non-names variables), and the non-names part (and no names variables). The names extract are encrypted and stored away in a safe and quite separate location and never used in any of the analyses.
11. Future developments in record linkage

11.1 Use of non-invasive physical characteristics of a person

Each individual has anatomical and physiological characteristics that are unique to him/her, such as fingerprints, eye colour, voice graph etc. and could potentially be used for matching. However, in medical practice, methods to record such characteristics are usually ethically unacceptable and often therapeutically undesirable. Over the past few years much work has been undertaken on the recording of fingerprints, facial features, facial heat patterns, retinal patterns, and gait. Since the cost of large-scale on-line computing is becoming much less expensive, it is now possible to store these identifying patterns in a person's personal record. The banking industry have developed the retinal pattern identification methodologies to supplement the plastic card for the money dispensing activities in the high street, and the system is undergoing extensive field testing. Devices are also available that can be clipped to the computer keyboard for reading fingerprints which can be used in addition to the normal password. These methods will provide quick, easy and accurate means of correctly identifying a person. Other methods would include the identification of a person through their DNA, but how this would be accepted by the general public and undertaken is not known.

11.2 Upgrade of the parsing, editing and blocking routines

The routines used for parsing the matching variables are being refined, since very large arrays of synonyms can be stored and accessed very efficiently by modern computing systems. Some work is being undertaken on the source of names in support of the spelling and other corrective routines.

11.3 Matching software based on rule-based search methods being developed for the Internet

Much development is being undertaken in the use of normal spoken language to initiate searches on the Internet. Search engines parse the enquiry into the appropriate keywords and start the search. The responses are graded according to how well they fit the enquiry. This methodology could be adapted to the matching process. Automatic learning and information retrieval can be used to develop a vocabulary or index that can be used in the comparison of records. Commonly occurring words such as 'the' can be removed in the parsing procedures and word stems could be stored and used to retrieve all variations on the word, e.g., medic for medical, medicine and medication. This extra level of parsing and structure would assist the record matching software in choosing or being restricted to which words to compare. From expert experience, such lists can be readily assembled and updated. (Nigam, 2000)

11.4 Use of intelligent and expert methods

These methods can be used to determine the matching thresholds over a wide range of matching conditions thereby improving the accuracy of the match and reducing the amount of clerical scrutiny. Since these methods can be tuned from running many samples, it might be possible to reduce the clerical steps substantially, thereby reducing the costs of the process, improve the quality of the match and reduce the overall elapsed time.
12. Ethics, confidentiality and data protection issues in record linkage

Anyone considering data linkage should be aware that there are ethical and legal issues involved with using data on individuals and businesses and these need to be carefully considered at the outset. This report is concerned with linkage for statistical purposes only, to produce summaries necessary for policy formulation and monitoring. Linkage of datasets may occasionally be for purposes other than that for which the data were originally collected and it is essential that the propriety of the new use be established before any linkage is attempted.

A further issue with record linkage is that although individual anonymity (of persons or businesses) may be protected in the source datasets, the result of linkage may be to provide a combination of attributes from which identity may be deduced. The requirements for confidentiality also include provision of a controlled, secure physical environment for the computer processing system. Control must also extend to paper listings produced in the matching or subsequent analytical processes.

The matching of datasets should include consideration of the following:

1) The ethics of linking the datasets, depending upon the source and content of the datasets.

2) Confidentiality of data about individuals and businesses in the linked result set.

3) Physical safeguarding of confidentiality including security of computer systems and administrative systems.

4) In the case of data relating to living individuals and to businesses, compliance with the relevant legislation.

Guidance on types 1, 2 and 3 of these issues will be found in the forthcoming National Statistics Code of Practice and associated protocols. Any matching for statistical purposes should take account of this code as a basis for good practice. Primary legislation referred to in (4) includes the Data Protection Act 1998 covering living individuals, and the Statistics of Trade Act 1947 covering businesses. The use of some individual datasets may be the subject of other specific legislation – there are some 200 bits of legislation covering the confidentiality of data.
13. Conclusions and recommendations

The role of data sharing and matching in the production of official statistics is likely to increase in the future. It has the potential to improve the quality of existing outputs and at the same time may lead to new and complementary outputs. Linkage of data sources is cited in the Cabinet Office ‘Modernising Government’ publication Adding It Up as important in deriving the ‘big picture’. We should make efforts to improve matching methodology during the design, data collection and record linking phases, and use data files that are representative of the whole population in order to produce high quality statistics for government decision-making.

The Task Force makes the following recommendations:

1) National Statistics experience of implementing automatic record linkage is limited. The development of a central resource and methodological capabilities for data linkage would be valuable.

2) The theory and methodology of record linkage is now well established. Deeper and thorough understanding of the practical issues involved in record linkage, such as selection of the matching variables, parsing and editing routines, calculation of weights, can only be obtained through ‘Hands-on’ experience. It is difficult to visualise in an abstract way how the identifiers used in matching can be compared and measured, and in turn how those measurements can designate a match, no match or possible match. During the survey of matching application in GSS, a number of possible projects with strong elements of probabilistic matching were identified. Joint working between the departments with business need for matching and the proposed central resource would be desirable.

3) Software packages currently available for undertaking record linkage have been described in this report. These packages are generally expensive and require substantial amount of training to use. It would be helpful to evaluate the available packages from the point of view of implementing the best one in National Statistics, or alternatively developing specific in-house National Statistics software.

4) The analysis of linked data sources, for example longitudinal data on individuals or businesses, is generally seen to be far more difficult and challenging than the analysis of individual data sources. The linked data sources are often spatially and temporally correlated. New techniques have been developed for the analysis of such data but these are largely inaccessible through lack of the necessary expertise. Acquiring skills and expertise in the use of these techniques is seen to be crucial to the linkage and exploitation of the hidden potential of administrative data sources.

5) Further research in record linkage methodology should be also undertaken, involving experts in universities and other National Statistics Institutes such as the US Census Bureau. The goal should be to overcome some of the present problems in the accuracy and applicability of record matching for the benefit of: research, healthcare and education provision, analysis and dissemination of results, funding and regulatory agencies, ethics and consumers of information. This will require some innovative applications of computer science, statistics and just plain common sense to emulate the actions of a very experienced, highly trained clerk.
Annex A Felligi-Sunter theory of probabilistic record matching and linkage

Felligi-Sunter theory of probabilistic record matching and linkage

The Felligi-Sunter statistical model starts with two files of records, A and B. The object of record linkage is to recognise the records in the two files which represent the same person (or other kinds of entities). All possible pairs of records, one from each file, are examined. These pairs of records \((a, b)\) are called comparison pairs. To be a match, the comparison pair \((a, b)\) must consist of records for the same person or entity. Accordingly, a pair consisting of records for two different persons or entities is a non-match or an unmatched pair.

For each comparison pair, one of three linkage decisions, \(D_{i}\), for \(i = 1,2,3\), is to be made. The decisions are:

\[
D_{1} - a \text{ and } b \text{ are for the same person or unit (called a positive link).}
\]

\[
D_{2} - a \text{ decision is not possible without further investigation (called a possible link).}
\]

\[
D_{3} - a \text{ and } b \text{ are for different persons or units (called a positive non-link, or a nonmatch).}
\]

The linkage decision taken for each comparison pair depends on rules based on the extent of agreement observed between the values of the matching variables for records \(a\) and \(b\). In the Felligi-Sunter model, the decisions are based on ratios of conditional probabilities: the probability of the observed result of the comparison, given that \(a\) and \(b\) are in fact for the same person or unit and the probability of the same observed result, given that \(a\) and \(b\) represent different persons or units.

Decisions may, of course, be incorrect. A false match occurs when the decision \(D_{1}\) is made and \(a\) and \(b\) represent different persons or units. A false non-match occurs when the decision \(D_{3}\) is taken and \(a\) and \(b\) represent the same person or unit. In the Felligi-Sunter model, the respective probabilities of false matches and false non-matches are denoted by \(\mu\) (type II error) and \(\lambda\) (type I error). Different decisions will have different costs associated with them. Operational costs are likely to be highest for the decision \(D_{1}\), which requires further clerical investigation to make a positive determination.

A linkage rule associates one of the three decisions \(D_{i}\) with every possible result of observing the values of the matching variables for a pair of records \((a, b)\). In the Felligi-Sunter model, an optimum linkage rule is one which achieves specified values of \(\mu\) and \(\lambda\), and minimises the number of pairs classified as possible links (decision \(D_{2}\)).

As Felllegi and Sunter (1969) point out, their theory for record linkage could have been formulated in terms of the classical theory of hypothesis testing, with their form of linkage rule being equivalent to a likelihood ratio test and their optimum linkage rules being the uniformly most powerful test for the alternative null hypotheses of the pair \((a, b)\) being a non-match or a match. Kirkendall (1985) has shown that the test statistic and optimum linkage rule used in their model can also be derived by using an information theoretical approach (Kullback, 1968).

While their basic model calls for the comparison of all possible pairs \((a, b)\) formed by elements of files A and B, Felllegi and Sunter also consider the possibility of file blocking, i.e. restricting the comparisons to pairs for which \(a\) and \(b\) are in agreement for one or more matching variables. Despite the enormous power of the present generation of computers,
examination of all possible pairs \((a, b)\) is still not economically feasible, even for medium-size files, so some form of file blocking must be employed. The most likely effect of blocking is to decrease the probability of false matches and increase the probability of false non-matches or missed matches. The first of these outcomes is, of course, desirable; the second is not. Fellegi and Sunter examined these effects and discussed methods of choosing among alternative blocking procedures. Kelley (1984, 1985) provides further guidance on how to make an objective choice among alternative blocking procedures by weighing the reduced costs of computation against the errors introduced by not looking at all comparison pairs.

As indicated in the paper from Tepping (1968), one of the key questions in using model-based record-linkage systems is how to estimate the weights. In the Fellegi-Sunter model, the problem is to estimate the likelihood ratios, often called weights, associated with the possible outcomes of the comparisons. Fellegi and Sunter describe two methods for estimating weights. The first method assumes the availability of prior information on the distribution of the matching variables in the populations from which files \(A\) and \(B\) are drawn, as well as on the probabilities of errors in generating the individual records that are compared. The method of calculating the outcome weights is explained in Section 9.3.

The second method, which requires an assumption that errors in recording different matching variables be independent, estimates the components of the weights directly from the frequency of the matching variables in the files being linked.

Under the Fellegi-Sunter model, each comparison pair is examined and classified into one of the three decision categories independently of all other pairs. Consequently, a record in file \(A\) can be classified as a positive link with two or more records in file \(B\) and vice versa. This may or may not be a satisfactory outcome, depending on the objectives of the record linkage and on what is known or assumed about the possibility of duplication in either of the input files. The appearance of groups of linked comparison pairs with common elements must be dealt with in practical record linkage applications. Howe and Lindsay (1981) and Kirkendall (1985) discuss methods that are appropriate, given various assumptions about duplication in the input files. In the matching system being developed by the US Bureau of the Census, the initial assignment of positive links is done by an optimisation process that does not permit any multiple linkages. Subsequently, however, the weights of other comparison pairs involving the linked records are systematically reviewed to identify possible duplicates (Jaro, 1985).

**Extension to the Fellegi-Sunter theory proposed by Tepping**

The Tepping (1968) model for record linkage uses the same underlying framework as the Fellegi-Sunter model, but differs in some important ways. The Fellegi-Sunter model permits only three possible decisions for each comparison pair: positive link, possible link and positive non-link, and in some applications the possible link category is dispensed with (e.g., Howe and Lindsay, 1981). Tepping's model does not restrict the number of possible decisions that can be taken for a comparison pair. He gives an example with five alternatives which could be characterised as positive link, positive non-link, tentative link, non-link, and possible link.

Fellegi and Sunter point out that costs or losses associated with each of the possible decisions can be taken into account in setting the error levels \(\mu\) and \(\lambda\). Tepping, however, makes costs an explicit element of his model. Costs include both operational costs of record linkage and those losses associated with the clerical resolution of matching errors. For each decision \(D(i)\), cost is assumed to be a function of the conditional probability of a match, \(P(\text{Match} \mid \gamma)\) where \(\gamma\) is the outcome of the comparison for a pair. The linkage rule is simple: for any specific value of \(P(\text{Match} \mid \gamma)\), choose the decision \(D(i)\) that has the smallest cost. This rule clearly minimises the overall costs and is therefore an optimum linkage rule.
The parameters that must be estimated to apply the Tepping model differ from those needed for the Fellegi-Sunter model. The parameters include both the cost functions for the decisions $D(i)$ and the values of $P(\text{Match} \mid \gamma)$ for each possible comparison outcome. To assign the values of $P(\text{Match} \mid \gamma)$, one needs to estimate what proportion of the comparison pairs with each possible outcome are, in fact, matches. Tepping proposes that this be done by taking a sample of pairs for each outcome and attempting to determine their true match status using clerical intervention, presumably on the basis of additional information obtained for these cases.

The two models have many elements in common and both provide useful guidelines for practical record-linkage operations. An important consideration in choosing between them would be a judgement of the feasibility of estimating the necessary parameters that, as we have seen, are different for the two models.

A variation of Newcombe’s ideas was later mathematically formalised by Fellegi and Sunter (1969) (see Section 7.2). (see also Winkler, 1988, 1989c for extensions). Copas and Hilton (1990) introduced a new theoretical approach that, in special cases: has some aspects of Newcombe's approach, but it has not yet been applied to any large scale record linkage system.
Annex B  Phonetic coding systems

Phonetic coding systems are employed most frequently to bring together variant spellings of what are essentially the same names. For this purpose, the files may be sequenced by the code, and the comparisons are then carried out only between records having the same codes. Occasionally, phonetic coding is used instead to reveal similarities between names even where the codes themselves may differ. A third use of such codes is simply to reduce the sizes of the names, although this is rarely undertaken.

Only the most commonly employed phonetic coding systems are discussed here. Most such systems have two features in common:

1) The vowel information is either partially or wholly suppressed because of its instability.

2) Certain consonants with similar sounds (or groups of letters containing these consonants) are replaced by a standard character (or group of characters) representing that sound.

The NYSIIS, Soundex and other variants possess these features. However, the NYSIIS retains information on the sequence of vowels in the name by changing them all to the letter 'A', whereas the Soundex gets rid of the vowels. 'Name compression' and the 'ill-spelled name routine' resemble the Soundex in this respect.

All of the above codes are capable of revealing similarities between names even where the coded forms do not agree precisely. However, only the ill-spelled name routine has been systematically applied to distinguish different degrees of similarity where comparisons between names are carried out freely.

Finally, where shortened forms of the names are needed for compactness, name compression may be desirable because it loses only the vowels and the redundant consonants.

a) The Soundex code

The Soundex code is particularly efficient at reducing more of the unreliable components of a name than does the NYSIIS algorithm, but it also loses more of the available discriminating power in the process. This is partly because it discards information on the positions of the vowels in the name.

The Soundex code is widely used in medical record systems despite its disadvantages. Although the algorithm copes well with Anglo-Saxon and European names, it fails to bring together some common variations of names, such as THOMSON/THOMPSON, HORTON/HAWTON, GOFF/GOUGH etc., and it does not perform well where: the names are short, such as those of the very common names (e.g. BROWN, SMITH, EVANS, GREEN etc), or where the names have a high percentage of vowels, or are of Oriental origin (e.g. LO, LOW, AU, LEE).

There are many variations of the Soundex code, for example Miracode, SINGS and Nu-Soundex. The Soundzee and the SINGS algorithms were developed to circumvent the problems with the retention of the first letter of the name (e.g. problems with YEAGER and JAEGER) where the first character of the code is a number prepared in the same manner as the other equivalence classes, and extended to cover the letters AEIOU and WHY which are normally deleted by the Soundex algorithm.
The steps in the coding procedure are simpler than for the other algorithms. In most variants of Soundex the first letter of the name is retained as the first letter of the code. The vowels are removed, consonants are assigned class numbers from 1 to 6 to represent their sounds, and redundant code numbers are removed.

The detailed rules for the original Soundex algorithm are:

1. The first letter of the name is used in its uncoded form to serve as the first character of the code. (The rest of the code is numerical).

2. Thereafter, W and H are ignored entirely.

3. A, E, I, O, U, Y are not assigned a code number, but serve as 'separators' (see Step 5).

4. Other letters of the name are converted to a numerical equivalence classes:

   B,P,F,V ➔ 1
   C,G,J,K,Q,S,X,Z ➔ 2
   D,T ➔ 3
   L ➔ 4
   MN ➔ 5
   R ➔ 6

5. Prefixes (va, Von, Di, de, le, du, d', dela etc) are sometimes removed or converted by the parsing routines (see Section 5.2).

6. There are two further reduction processes:

   (a) letters that follow prefix letters which would, if coded, have the same numerical code, are ignored in all cases unless a 'separator' (see Step 3) precedes them.

   b) The second letter of any pair of consonants having the same code number is likewise ignored, unless there is a 'separator' between them in the name.

7. The final Soundex code consists of the prefix letter plus three numerical characters. Longer codes are truncated to this length, and shorter codes are extended to it by padding out the code with zeros.

Examples:

ANDERSON, ANSERSEN ➔ A536
BERGMANS, BRIGHAM ➔ B625
BIRK, BERQUE, BRICK ➔ B620
FISHER, FISCHER ➔ F260
SMITH, SMYTH, SNEATH, SNAITH, SMOOTHEY, SAMUDA ➔ S530
LLEWELLYN ➔ L450
AU ➔ A000
GILL ➔ G400

b) Daitch-Mokotof code

Gary Mokotoff computerised the 1984 list of names of persons who changed their names while living in Palestine under the British mandate. the Soundex did not work well with Slavic and Germanic spellings of Yiddish surnames. He proposed the revised Jewish Soundex Code and it is known as the Daitch-Mokotoff code.
Rules for the Daitch-Mokotoff extensions to the Soundex Code

1. Expansion of the Soundex code from 4 to 6 digits
2. Code the first letter
3. Assign a single letter to some double letter combinations
4. Code more than once, letters/combinations pronounced in different ways
5. Changed Romance language pronunciations to Slavic/German
   the letters W to V are not silent
6. Addressed the problem of letter C, the hard and soft sound are separated

\[ \text{c) The New York State Identification and Intelligence System (NYSIIS) code} \]

The NYSIIS coding algorithm is an extension of the Soundex algorithm and removes silent, redundant or duplicate consecutive letters. The resulting code is totally alphabetic. Since the vowels are retained, the NYSIIS blocks are smaller than those produced by Soundex, and contain less dissimilar surnames. The algorithm progresses in three well-defined steps:

1) In the pre-fix stage, the first letter(s) of a name are tested and changed as necessary, for example:
   KN \( \Rightarrow \) NN (KNIGHT \( \Rightarrow \) NNIGHT) , and WR \( \Rightarrow \) RR. (WRIGHT \( \Rightarrow \) RRIGHT)

2) In the post-fix stage, similar conversion processes are carried out on the last letter(s)
   of the name, for example,
   DIX \( \Rightarrow \) DICKS, and LEAS, LEE \( \Rightarrow \) LEA

3) Lastly, in the in-fix stage, starting with the second letter of the text string, a scan is
   carried out of each letter of the name using the letter position as an imaginary
   'pointer'. At each stage, changes may be made to the name according to a recursive
   set of rules shown below. The recoding rules are applied in a recursive manner each
   time the 'pointer' is moved on to the next letter of the name.

Rules:

The rules are applied in a recursive manner each time the 'pointer' is moved on to the
next letter of the name. During the process, characters may be taken from the altered
name and transferred to create the NYSIIS code.

Finally, the end portion of the NYSIIS code is examined and subjected to further tests
and changed where necessary.

Usually the NYSIIS code for a surname is based on a maximum of twelve letters of
the full alphabetical name, and the NYSIIS code itself is then limited to six letters.
The detailed rules are:

1. Change the first letter(s) of the name.
   If xx replace with yy as follows:
   (a) MAC \( \Rightarrow \) MCC;
   (b) KN \( \Rightarrow \) NN;
   (c) K \( \Rightarrow \) C;
   (d) PH \( \Rightarrow \) PP;
   (e) PP--\( \Rightarrow \) PP;
2. Change the last letter(s) of the name.
   If xx replace with yy as follows:
   (a) S ➔ delete;
   (b) Z ➔ delete;
   (c) EE ➔ Y;
   (d) IE ➔ Y;
   (e) DT ➔ D;
   (f) RT ➔ D;
   (g) RD ➔ D.

3. Create the first character of the NYSIIS code
   First character NYSIIS code = the current first letter of the name.

4. Set the 'pointer' to the second letter of the name.

5. Change the current letter(s) of the name, (i.e. at the present position of the 'pointer').
   If xx replace with yy; execute one only of the following, in the order given:
   (a) blank ➔ go to rule 7;
   (b) EV ➔ AF;
   (c) vowel (AEIOU) ➔ A;
   (d) Q ➔ G;
   (e) Z ➔ S;
   (f) M ➔ N;
   (g) KN ➔ N;
   (h) K ➔ C;
   (i) SCH ➔ SSS;
   (j) PH ➔ FF
   (k) H preceded by or followed by a non-vowel (AEIOU) ➔ replace the current position in the name with the preceding letter;
   (l) W preceded by a vowel ➔ replace the current position in the name with the preceding letter;
   (m) if none of these rules applies; ➔ then retain the current position letter in the name.

6. If the current position letter in the name is equal to the last character placed in the NYSIIS code, do not enter it into the code. Instead, set the 'pointer' to the next letter of the name, and go to Step 5.

   The next character of the NYSIIS code = the current position letter in the name after completing Step 5 (but omitting a letter that is equal to the last character already placed in the code).

   After putting a character into the code, move the pointer forward to the next letter of the name. Then go to Step 5.

7. Change the last character(s) of the NYSIIS code.
   If xx replace as follows:
   (a) S ➔ delete;
   (b) A Y ➔ Y;
   (c) A ➔ delete.
There are many local modifications of the NYSIIS name-coding procedure which follows the above pattern but includes additional rules. These are summarised under the numbers and headings used above, as continuations of the lists of codings already given.

1. **Change the first letter(s) of the name.**
   If xx replace with yy as follows:
   (a) WR → RR;
   (b) RH → RR;
   (c) DG → GG;
   (d) vowels (AEIOU) → A.

2. **Change the last letter(s) of the name.**
   If xx replace with yy as follows:
   drop terminal S or Z from all names before coding begins;
   (a) YE → Y;
   (b) NP → N;
   (c) ND → N;
   (d) IX → IC;
   (e) EX → EC;
   (f) JR → blank; (for Junior)
   (g) SR → blank. (for Senior)

**Change the current letter(s) of the name, i.e.**
 at the present position of the 'pointer'. If xx replace with yy as follows:

   (a) Y when not the last letter of the name → A;
   (b) SCH at end of the name → SSA;
   (c) SCH not at end of the name → SSS;
   (d) SH at end of the name → SA;
   (e) SH not at end of the name → SS;
   (f) GHT → TTT;
   (g) DG → GG;
   (h) WR → RR.

3. **Change the first character of the NYSIIS code.**
   If xx replace with yy as follows:
   (a) A → first letter of the original name;
   (b) space → first letter of the original name.

These modifications to the original NYSIIS code are designed to deal with rare or difficult cases. Thus, most names will have identical codes whichever version of the coding procedure is used.

**Examples:**

ANDERSEN, ANDERSON → ANDAR
BRIAN, BROWN, BRUN → BRAN
CAPP, COPE, CAPP, KIPP → CAP
DANE, DEAN, DENT, DIONNE → DAN
SMITH, SCHMIT, SCHMIDT → SNAT
TRUMAN, TRUeman → TRANAN
UHCE extensions
LEE, LEES, LEA, LEAS, LEIGH → LA

d) Oxford Name Compression Algorithm (ONCA)
The Oxford Name Compression Algorithm (ONCA), uses a version of the NYSIIS method of compression as the initial or pre-processing stage, and the transformed and partially compressed name is then Soundexed in the usual way. This two-stage technique has been used successfully for blocking the files of the ORLS, and overcomes most of the unsatisfactory features of pure Soundexing while retaining a convenient four-character fixed-length format. (Gill et al, 1982, 1987, 1993, 1997). The ONCA code is presented in Box B.1

<table>
<thead>
<tr>
<th>Original name</th>
<th>NYSIIS Code</th>
<th>ONCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANDERSEN, ANDERSON</td>
<td>ANDAR</td>
<td>A536</td>
</tr>
<tr>
<td>BRIAN, BROWN, BRUN</td>
<td>BRAN</td>
<td>B650</td>
</tr>
<tr>
<td>CAPP, COPE, COPP, KIPP</td>
<td>CAP</td>
<td>C100</td>
</tr>
<tr>
<td>DANE, DEAN, DENT, DIONNE</td>
<td>DAN</td>
<td>D500</td>
</tr>
<tr>
<td>SMITH, SCHMIT, SCHMIDT</td>
<td>SNAT</td>
<td>S530</td>
</tr>
<tr>
<td>TRUMAN, TRUEMAN</td>
<td>TRANAN</td>
<td>T655</td>
</tr>
</tbody>
</table>

The blocks produced using ONCA alone vary in size, from quite small and manageable for the less-common surnames, to very large and uneconomic for the more common surnames. Further sub-division of the ONCA blocks on the file can be effected using sex, or forename initial, and date of birth either singly or in combination.

e) Name compression and the ill-spelled name routine
'Name compression' is a way of reducing the length of a name; in addition, it serves as a preliminary step in the 'ill-spelled name routine'. For this latter reason the two are considered together.

Rules:
1. Delete the second of any pair of identical consonants.
2. Delete A, E, I, O, U, Y, except when the first letter of the name.

Examples:

BENNETT → BNT
FISHER → FSHR

f) Ill-spelled name routine
This routine was designed to avoid a limitation inherent in other systems of coding and comparing names. The manner in which the NYSIIS and Soundex codes are commonly used, detailed comparison of the full alphabetical versions of a name is dependent on exact agreement of the phonetic codes; other degrees of similarity of the codes are not taken into
account. This is true even where the codes differ with respect to just the simple insertion or deletion of a single character in the one as compared with the other.

An 'insertion' into, or a 'deletion' from, any sequenced string of characters results in what is called a 'frame shift' relative to the previous sequence. This will affect the comparison of the one version with respect to the other. If the part of the sequence on one side of the insertion/deletion point is seen to be in perfect agreement, the other part may appear totally different, and vice versa. It is all a matter of where one starts reading. The main problem is to design a comparison system that will detect similarities in spite of such frame shifts. A secondary problem is to make the system work even where there is a different insertion into each of the two sequences. The ill-spelled name routine is intended to solve both of these.

**Rules:**

1) Use the name compression procedure, up to a total of four letters.

2) When comparing two such four-letter codes, search for and count the numbers of letters or blanks, up to a total of four in all, that agree without altering their sequence (i.e. skip over any inserted letters that would otherwise interrupt the sequence that agrees).

3) Any of five levels of similarity/dissimilarity may be revealed by such a comparison (i.e. 0, 1, 2, 3, or 4 agree). This 'score' may be used in various ways. For an automated linkage operation, the FREQUENCY RATIOS in LINKABLE versus UNLINKABLE pairs would be obtained for each of the five levels of outcome, and used in calculating the ODDS in favour of a correct link. For the purposes of manual resolution, a score of 3 or 4 might indicate a need for visual comparison of the full alphabetical versions of the names.

**Examples:**

- BOWMAN/BAUMAN ➞ BMN/BMN ➞ 4(BMN-)
- ANGRIEF/SINGER ➞ ANGR/SNGR ➞ 3(NGR)
- LEICESTER/LESTER ➞ LCST/LSTR ➞ 3(LST)

**g) Name compression algorithm**

As indicated by its name, this form of coding is designed mainly to condense surnames, given names and place names. However, the code does remain unchanged with some of the common spelling variations, although it is less efficient in this respect than the Soundex code.

**Rules:**

1) Delete the second of any pair of identical consonants.

2) Delete A, E, I, O, U, and Y except when the first letter of the name
Examples:

BENNETT  →  BNT
FISHER  →  FSHR

h) Alphanumeric conversion algorithm

This is a highly specific numeric coding for all surnames. It is not designed to set aside the less stable parts of the information but rather to retain virtually all of the original specificity of the alphabetic form. The numeric form of the surname is compact, is more readily sorted, and is non-revealing to anyone who lacks the relevant look-up table. Furthermore, when sorted into numerical sequence the names fall into alphabetical order or close approximation to it. The coding is done by computer using a look-up table of 10,000 or more entries.

Examples:

ABBIT  →  0008
ADLER  →  0105
BORNE  →  1058
BRYAN  →  1070
CLARK  →  1646
COX  →  1721
.........
ZZINA  →  9776

i) Davidson code

The Davidson code is a way of reducing the length of a name; in addition, it serves as a preliminary step in the 'ill-spelled name routine'.

Rules:

1) Delete A, E, 1.0, U, W, H, Y, except when the first letter of the name.
2) Delete the second of any pair of identical consonants.
3) If the resulting code is less than 4 characters long, add spaces to the end of the code to make 4 characters in all: if it is more than 4 characters, keep only the first 4.
4) To the end of the code add the first letter of the first surname

Examples:

BENNETT. J  →  BNT\nS (where \n = space)
FISHER. S  →  FSHRS
Annex C Other record matching algorithms

The two main matching methods, Exact Matching and Probabilistics Matching, have already been described in detail in Sections 6 and 7. To exploit the developments in tools and technologies, many other matching methods have been developed. A number of these methods are rule based, and support internet search engines and the World Wide Web. The following list, while non-exhaustive, is a guide to the methods that have been developed:

a) Brute force 1:1 exact matching

It is normal practice in exact matching to compare only those records that have identical keys. Therein lies the simplicity of the method. However, where a compound identifier has been assembled from a small number of reliable identifiers, (e.g. date of birth, gender and postcode) there is the possibility of error or omission in any of the components, it is then desirable to match the query against all the possible candidate records on the master file. The Brute Force matching approach is to match a query string with all the records that contain the query string or parts of query string. This is similar in concept to exact matching with the proviso that every record is matched with all the other records on the file, and in each record pair every character in the query string is matched with the corresponding character on the master record. For small files this process can be efficient and inexpensive, but for large files, say, of the order of 1 million records the number of comparisons will be $5 \times 10^6$ and even at 1 million comparisons per minute would take roughly 347 days to complete using a desk top computing system.

b) Rule-based matching systems

Rule-based or Boolean matching systems arose when the IT technology used punched cards and edge-notched cards. With this technology, separating the records containing a given term from those that do not is the most obvious methodology. This operation translates easily into the logic of Boolean retrieval. In such systems, the queries are sets of Boolean statements using the AND, OR and NOT operators, and are usually of a different structure from the textual records on the master file. Each query, is a logical function prepared in the form of a Boolean statement; a document, in the conventional sense, is not defined in this way. Since there is no structural similarity between record and query, the query is regarded as a set of logical instructions, and retrieval with respect to a given query is regarded as a characteristic function defined on the record space. Since the process will only return those records that match with the Boolean statement, their advantages in retrieval systems have until lately have been of limited use (Verhoeoff, Goffman, and Belzer, 1961).

Matching can then be viewed as a mapping of the Boolean rules in the query record with all the possible master file records. Each record is mapped, or transformed, into a representation compatible with that of the query. The system then determines whether the transformed record meets the requirements of the query.

A purely rule-based Boolean retrieval system provides no basis for the development of significant similarity judgements. By definition, a given record either satisfies the Boolean query or it does not. Since the mapping defined by a query is a characteristic function, it divides the documents into those that satisfy the query and those that do not.

Various modifications of Boolean query systems permit some finer grading of the set of retrieved records. Consider, for example, the query $A \ OR \ B \ OR \ C$. This is satisfied by any record containing at least one of the three terms. Some of these records will contain only one of the terms, while others will contain two or all three. Thus the retrieved set can be graded by how many of the three terms each record contains and even by the specific terms, thus separating the records with the terms $A$ and $B$ but not $C$ from those with the terms $A$ and $C$ but
not B, and both of these sets from the records that contain all three terms. If the system permits proximity judgements then gradations of the retrieved set can be made on that basis.

Since Boolean systems operate on the basis of the presence or absence of terms, many such systems do not include the term frequency data. In this case organisation of the retrieved set on the basis of similarity measures depending on frequency cannot be performed.

Rule based systems are used quite extensively for matching query strings in the World Wide Web (Internet) search engines being developed and in current use

c) Partial matching methods and the use of wild card characters

A related matching problem is that involving wild-card, symbols which match any single symbol including another wild-card symbol.

The characteristics of the 'partial-match' or 'crossword puzzle' is that the string may contain both regular letters and the wild-card character '. Searching the dictionary for the pattern 'u-u-u' matches the single word ahuulu, while the pattern 'a-a-a' matches 94 words, including banana, casaba and pajama. (That pattern does not match abracadabra, though). The two strings must match the pattern including the position of the wild-card characters. A restriction can be applied for not searching for patterns in the middle of longer strings.)

A search of records containing the term medicine would not generally produce good matches with related terms such as: medical, medicinal and medication. It is possible to search for all such terms using the string replacement mechanisms that are offered by the commercial software vendors. For example, one could search for the term medic, which would return all those records that contain terms beginning with the string medic, since the ' ' operator replaces all the possible suffix strings. In addition, it may be possible to have a single character replacement operator for either prefixes or suffixes.

This method of string replacement is called wild card searches, characters that can be matched with anything. Another version of this concept has been used in many World Wide Web search engines where the '?' character is matched with any other single character in the same position in the text string. The de facto standard for wild cards in string matching is to use a question mark ('?') to denote a match by at most one character, and an asterisk ('*') to denote a match by an arbitrary number of characters. The matching problem is complicated by the fact that the match relation is no longer transitive, for example: AB matches with A?, and A? matches with AC, but AB does not match with AC. An example of this technique in record matching would be the match between ROBERTS and ROBERTSON, using the query ROBERT*

In reality, wild-card methods only work if the researcher correctly guesses the right characters to include or exclude. The method can be used to find all the relevant records which contain the stem string, but cannot be used where the dataset contains variations on the original strings, for example synonyms and abbreviations. Wild-card searches will always return too many irrelevant records and cannot be considered for all but the simplest of searching methods.

d) Proximity or near-neighbour matching

The objective of 'near neighbour' searching in a set of strings is to find all strings in the file that are within a given distance of the query string. For example, a search for all strings within distance two of Dobbs finds Debby, hobby, etc.. The algorithm is quite efficient when searching for near neighbours, but searching for distant neighbours grows more expensive. A good example of this type of matching is used when matching date of birth from many document sources, where the date may be discrepant by a number of days, months or years.
Modifications of proximity criteria can increase their effectiveness. One such modification is to use phrases rather than simple word proximity. The incorporation of phrases into the judgement process requires that a given set of words occurs in a highly specific sequence. This focuses the search criteria more sharply but has the negative effect of discounting expressions that are equivalent to, but different from, the specified phrase. For example, if the phrase information retrieval is required, then a record referring to the retrieval of information will be rejected. A solution to this difficulty is to include standard proximity criteria but give added weight to records having the specified phrase.

Another modification, less strict than the use of phrases, is the use of ordered proximity to aid in the retrieval decision. Very clearly, the order in which words occur affects their interpretation. One classic example with a clear distinction in interpretation is junior college versus college junior. Although ordered proximity will distinguish between these two phrases, selecting only records with the desired interpretation, it will sometimes cause some relevant records to be missed. A decision to use either phrase or ordered proximity techniques depends on knowledge not generally available to a computer system and thus must be made by the user or a trained information professional.

e) Fuzzy matching

The problem with probabilistic matching is, of course, in estimating the weights and calculating the probabilities. Fuzzy matching appears similar to probabilistic matching but replaces the need to estimate probabilities by a need to estimate if the record is relevant. Thus, the strict rules that govern the use of probabilities do not apply. A fuzzy judgement can be made about whether a record should be in the set which matches with the query. This is done on the basis of a set of terms describing the record or the variables used in the record.

Fuzzy systems are based on fuzzy set theory and associated techniques pioneered by Lotfi Zadeh (1983). The aim of this approach is to mimic an aspect of human reasoning by doing approximate reasoning. In this way, fuzzy systems are less precise than conventional systems but are more like our everyday experience as human decision makers. Fuzzy terms are not precise but they are meaningful, and they allow us to describe entities and reason about them.

Recall, that in probabilistic matching the computation ultimately devolved into the calculation of the probabilities that terms in the data record are selected as potentially relevant to the corresponding terms in the master file records. In fuzzy matching the calculations are based on the terms being in the same set, such as: Berkshire, age 47, SMITH etc. The question then becomes one of the degree of confidence that a record containing a given term is relevant. Using the arithmetic of fuzzy sets, the match between the data record and the master file record can be computed. This computation is simpler than that for probabilistic retrieval, since it involves simple functions of the membership set for each data record.

Different approaches can be taken to the concept of fuzziness. In considering semantically related terms, one might consider how well a related term matches a given term. For example, if the query is about JANET SMITH, a record containing the term JEANETTE SMITH would not match, but the two people may be related closely enough that the record may still contain useful information. The match depends, of course, on the comparison of the forename Janet with Jeanette, a fuzzy judgement.

f) Use of Internet Search Engines for matching and retrieval

Search systems developed for the Internet and the World Wide Web (WWW) can be regarded as consisting of four functional modules: gatherer, indexer, search and retrieval. The gatherer traverses the documents on the Web as a spider traverses a spiders web. It collects information about the documents to be indexed. The gatherer starts from a single document, usually the home directory of the Web server, then selects the next document to index.
Documents on the Web are complex, widely distributed, and dynamic, so that it is impossible to gather every document on the Web. The indexer takes the information collected by the gatherer, creates index records and enters them into a database. In some systems the functions of the gatherer and indexer are combined into one module. Indexing is the key component of the search system; a high quality index leads to precise identification and retrieval of documents and can reduce the time required for searching. The search module contains two functions, a search language and a search engine. The search language transforms the user’s search query into a formal representation of a database query language; the search engine then applies the formal representation to the database, and extracts all the records that meet the specified query.

There are two popular types of search languages on the Web: keyword matching and rule-based or Boolean searching.

Keyword matching is a search language that can directly combine terms specified by the user into a search query. The user can specify any word or part of a word or use wildcard characters in the query to the search engine. Keyword matching works by pattern-matching the characters in the query to characters in the database records.

Rule-based or Boolean search is a more sophisticated search language that serves to narrow and refine keyword-matching searches. Boolean search language allows the user to combine keywords using the Boolean operators AND, OR and NOT. Where the search system supports Boolean search the user can input words, phrases or combinations of phrases as a search enquiry.

A variation on Boolean search is fuzzy Boolean, where the user provides a list of terms and the variables retrieved must match at least one of the terms. They are then ranked by the number of terms matched.

After gathering, indexing and searching are complete, the retrieval module gives the user access to the original documents located by searching, either by direct retrieval from the database or by retrieval from the original client-server.

**g) Expert systems and other intelligent software**

The understanding of natural language and its use in matching phrases is an extremely complex phenomenon. It involves the recognition of sounds, words and phrases and has applications in machine translation, encoding of phrases, and for the comparison of phrases as for example in matching and record linkage.

There are various levels in the process of language analysis:

**Phonetics** deals with the main sound units of speech (phonemes) and their correct combination. This type of analysis is used extensively in the development of phonetic compression algorithms

**Lexicology** deals with the words used in the language.

**Semantics** deals with the meaning of words and phrases seen as a function of the meaning of all the component parts.

**Syntax** deals with the rules which are applied to matching words in order to form sentences or in our case an identification set.

Humans, when communicating with one another share a lot of common sense knowledge which is inherited and learned in a natural way. This is a major problem for the development of computer programs that mimic the way in which humans can differentiate between patterns of information.
One way to represent inexact data, closer to human like thinking is to use fuzzy rules instead of exact rules when representing knowledge. Fuzzy systems are rule-based expert systems based on fuzzy rules and fuzzy inference. Fuzzy rules represent in a straightforward way knowledge that is subjective, ambiguous or vague. This knowledge might come from any of the input sources of data. Human clerks would apply common sense approaches to the resolution of the data matching based on long experience, or from the experience of many people, over many years.

During its development, expert systems have been moving towards new methods of knowledge representation and processing that are closer to human like reasoning. These systems are designed to provide reasoning similar to that of experts. A new computational method has already been established with many applications and developments, *artificial neural networks*.

The advantage in the use of neural networks is the manner in which they can be connected. The main characteristics of a neural networks are:

- **Learning** a network can start with 'no knowledge' and can be trained using a given set of data examples and produce desired outputs for known inputs.

- **Massive parallelism** during the processing of the data many aspects of the program will work simultaneously.

- **Robustness** if some of the processes go wrong the whole systems may still perform well.

- **Partial matching** is what is required in many cases since the existing data does not match well with the new facts.

These main characteristics of neural networks make them useful for knowledge engineering and for building expert systems. The systems can be 'trained' by inputting a set of examples. For example, where there are good clinical records about patients suffering from a particular disease, the network can be trained and to make logical output on the health status of the patients.

Usually neural networks are combined with other rule-based systems, fuzzy systems and probabilistic reasoning to provide a set of hybrid tools for coping with noisy data, missing data, imprecise or corrupted data and still provide a good solution.
Annex D Use of check characters for internal checking of numbering systems

Use of check characters

The use of unique person or entity numbering systems for record matching, though simple in concept, are prone to errors of recording, transcription and data preparation. It is therefore essential to consider methods of reducing the errors in their use. One such method is to incorporate check-digits or check-characters into this key matching variable (Wild, 1968; Hamming, 1986; Gallian, 1989; Gill and Baldwin, 1982; Sethi, 1978; Dass, 1984; Brown, 1973; Holmes, 1975).

To protect against such errors, decimal numbers such as those used in finance and business systems (for example, personal numbers or account numbers) have an extra character added to the sequence. There is usually some check algorithm which all the digits including the check digit itself must satisfy. In this way the composite number consisting of numeric sequence and check digit can be protected against a very high percentage of errors. If the check algorithm returns FALSE then there is something wrong with the numeric sequence. Conversely, if the check algorithm returns TRUE there may be one or more errors still present that are of a random type and cannot be detected by the check digit algorithm. The trick is to devise an algorithm that will test for, and trap, the known and expected errors that can arise from a given number sequence. A list of the errors that can arise in numbering systems is presented in Box D.1.

<table>
<thead>
<tr>
<th>Error</th>
<th>Explanation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single error</td>
<td>single digit entered in error</td>
<td>12345 becomes 12545</td>
</tr>
<tr>
<td>Insertion error</td>
<td>an extra single digit inserted</td>
<td>12345 becomes 123745</td>
</tr>
<tr>
<td>Deletion error</td>
<td>single digit omitted</td>
<td>12345 becomes 1345</td>
</tr>
<tr>
<td>Transposition error</td>
<td>two adjacent digits interchanged</td>
<td>12345 becomes 12435</td>
</tr>
<tr>
<td>Double error</td>
<td>two unrelated digits interchanged</td>
<td>12345 becomes 15342</td>
</tr>
<tr>
<td>Double transposition error</td>
<td>two adjacent digits changed ab→cd</td>
<td>12345 becomes 12675</td>
</tr>
<tr>
<td>Double repeated error</td>
<td>pair of equal digits changed</td>
<td>12335 becomes 12775</td>
</tr>
<tr>
<td>Random, multiple</td>
<td>Any combination of any of the above</td>
<td></td>
</tr>
<tr>
<td>errors</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shift errors</td>
<td>are insertion/deletion errors</td>
<td>12335 becomes 123335</td>
</tr>
</tbody>
</table>

As an example, the check algorithm may sum the individual digits in the number, divide by the modulus, and test the remainder to check if it equals the value of the check digit,

\[
(a_1 + a_2 + a_3 + a_4 + a_5 \ldots + \text{check digit}) \mod N = 0
\]

This method will always detect an error of a single digit, and will detect 90 per cent of all errors, but it will not detect the transposition of two or more digits. Where humans are involved the transposition of adjacent digits is a common trait so the algorithm is not ideal for use in exact matching. An extension of the above algorithm would be to use a different weight for each digit position in the number, for example,

\[
(1 \times a_1 + 7 \times a_2 + 3 \times a_3 + 5 \times a_4 + 2 \times a_5 \ldots + \text{check digit}) \mod N = 0
\]

This method of calculating a check digit is presented in Box D.3. An algorithm using modulus 11 will detect the transposition of two adjacent digits and 91 per cent of random errors,
whereas the use of modulus 97 will detect 99 per cent of random errors. The detection of some types of error is affected by the selection and arrangement of the weights, and ideally, both arithmetic and geometric progressions should be avoided, and it is important that no two adjacent weights should add up to the value of the modulus. A list of the percentage of errors detected using different moduli are presented in Box D.2.

<table>
<thead>
<tr>
<th>Modulus</th>
<th>Error Types</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Transcription/</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Substitution</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Transposition</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Single</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Double</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>93.3</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>100.0</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>100.0</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>100.0</td>
<td></td>
</tr>
<tr>
<td>97</td>
<td>100.0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Shift and Other</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>85.7</td>
<td>92.72</td>
</tr>
<tr>
<td>11</td>
<td>90.9</td>
<td>99.55</td>
</tr>
<tr>
<td>13</td>
<td>92.3</td>
<td>99.62</td>
</tr>
<tr>
<td>23</td>
<td>95.6</td>
<td>99.78</td>
</tr>
<tr>
<td>97</td>
<td>99.0</td>
<td>99.95</td>
</tr>
</tbody>
</table>

Source: Post Office, (1962)

The strength of the check digit algorithm is in the selection of a suitable modulus and the weights assigned to each digit position. In the majority of health related systems the modulus is normally base 11 and the weights are assigned from the left and ending on the right of the number. For example the new NHS number embodies a check digit with the modulus 11 and in which the weights are: 10,9,8,7,6,5,4,3,2. The method of calculating the check digit for the new NHS number is presented in Box D.3.

Alphabetic characters may be used for the check characters, however the number of characters that can be used are restricted to 23 since the letters 'O,' '1' and 'Z' can be confused with '0,' '1' and '2' respectively. Check digit systems using modulus 11 need additional restrictions since they require the use of a special symbol to represent the decimal number 10, which may be unacceptable in practice. Ten out of every 11 numbers will have check digits in the range 1 to 0, the eleventh number would generate a check digit of 10 which cannot be represented as a single digit and would require a non-numeric character to represent the 10, sometimes an asterisk, '*' is used. This is unacceptable where the number is issued to the public, and so every eleventh check digit number would never be issued.

More advanced check digit ing systems that use two or more check characters have been developed. These methods can be used to detect not only that an error has occurred but can also to determine which part of the number is in error. Extensions of this method has been used for many years to detect and correct 'in-flight' errors that occur in reading data from magnetic tapes and disks (Baylis, 1998).
Box D.3 Computation of the check digit in the new NHS number

The algorithm uses modulus 11 and weights 10, 9, 8, 7, 6, 5, 4, 3, 2
NHS Number 4 2 8 5 9 9 3 7 3 2 (the last digit is the check digit)
Weights 1 0 9 8 7 6 5 4 3 2
Cross Products 40 18 64 35 54 45 12 21 6 = 295
Divide by the modulus (11) = 295/11 = 24 remainder 9
The Check digit is calculated as (modulus - remainder) = 11 - 9 = 2
Annex E  Distribution of names in England and Wales 
from analysis of the NHSCR

Analysis of the names information that is stored on the NHS Central Register for England and Wales shows that there are:

57,963,992 records
1,071,603 different surnames
15,143,043 different surname/forename pairs

Source: NHSCR (1997)

The low frequency names were mainly non Anglo-Saxon names, hyphenated names or misspelled names. In general, the misspellings were due to embedded vowel changes or to miss-keying. A more detailed examination of the NHSCR showed that 954 different surnames described about 50 per cent of the population, these results are presented in Box E.1.

<table>
<thead>
<tr>
<th>Box 16.5 Frequency of surnames on the NHSCR</th>
</tr>
</thead>
<tbody>
<tr>
<td>24 different surnames covers 10% of the population</td>
</tr>
<tr>
<td>84 different surnames covers 20% of the population</td>
</tr>
<tr>
<td>213 different surnames covers 30% of the population</td>
</tr>
<tr>
<td>460 different surnames covers 40% of the population</td>
</tr>
<tr>
<td>954 different surnames covers 50% of the population</td>
</tr>
<tr>
<td>1,908 different surnames covers 60% of the population</td>
</tr>
<tr>
<td>3,912 different surnames covers 70% of the population</td>
</tr>
<tr>
<td>10,214 different surnames covers 80% of the population</td>
</tr>
<tr>
<td>100,000 different surnames covers 90% of the population</td>
</tr>
<tr>
<td>1,071,603 different surnames covers 100% of the population</td>
</tr>
</tbody>
</table>

Source: NHSCR (1997)
Distribution of names in England and Wales from analysis of the NHSCR: 
Frequency of top 20 surnames

**Box E.2 The top 20 surnames (both sexes all ages)**

<table>
<thead>
<tr>
<th>Surname</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMITH</td>
<td>708,620 (1,222.5)</td>
</tr>
<tr>
<td>JONES</td>
<td>579,412 (999.6)</td>
</tr>
<tr>
<td>WILLIAMS</td>
<td>402,002 (693.5)</td>
</tr>
<tr>
<td>TAYLOR</td>
<td>326,669 (563.6)</td>
</tr>
<tr>
<td>BROWN</td>
<td>313,580 (541.0)</td>
</tr>
<tr>
<td>DAVIES</td>
<td>299,017 (515.9)</td>
</tr>
<tr>
<td>EVANS</td>
<td>237,735 (410.1)</td>
</tr>
<tr>
<td>THOMAS</td>
<td>216,200 (373.0)</td>
</tr>
<tr>
<td>WILSON</td>
<td>213,868 (369.0)</td>
</tr>
<tr>
<td>JOHNSON</td>
<td>204,617 (353.0)</td>
</tr>
<tr>
<td>ROBERTS</td>
<td>199,440 (344.1)</td>
</tr>
<tr>
<td>ROBINSON</td>
<td>175,408 (302.6)</td>
</tr>
<tr>
<td>THOMPSON</td>
<td>172,257 (297.2)</td>
</tr>
<tr>
<td>WRIGHT</td>
<td>171,961 (296.7)</td>
</tr>
<tr>
<td>WALKER</td>
<td>165,926 (286.3)</td>
</tr>
<tr>
<td>WHITE</td>
<td>165,740 (285.9)</td>
</tr>
<tr>
<td>EDWARDS</td>
<td>162,569 (280.5)</td>
</tr>
<tr>
<td>HUGHES</td>
<td>160,627 (277.1)</td>
</tr>
<tr>
<td>GREEN</td>
<td>156,946 (270.1)</td>
</tr>
<tr>
<td>HALL</td>
<td>156,143 (269.4)</td>
</tr>
</tbody>
</table>

Note: The figures in brackets are the rate per 100,000.

Source: NHSCR, (1997)

**Box E.3 The top 20 forenames (all ages)**

<table>
<thead>
<tr>
<th>Male forenames</th>
<th>Frequency</th>
<th>Female forenames</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOHN</td>
<td>78,835 (272.0)</td>
<td>MARGARET</td>
<td>34,118 (117.7)</td>
</tr>
<tr>
<td>DAVID</td>
<td>70,888 (244.6)</td>
<td>SUSAN</td>
<td>33,138 (114.3)</td>
</tr>
<tr>
<td>MICHAEL</td>
<td>51,023 (176.1)</td>
<td>SARAH</td>
<td>24,951 (86.1)</td>
</tr>
<tr>
<td>PETER</td>
<td>41,387 (142.8)</td>
<td>MARY</td>
<td>24,471 (84.4)</td>
</tr>
<tr>
<td>JAMES</td>
<td>39,851 (137.5)</td>
<td>ELIZABETH</td>
<td>23,058 (79.6)</td>
</tr>
<tr>
<td>PAUL</td>
<td>39,730 (137.1)</td>
<td>PATRICIA</td>
<td>22,039 (76.0)</td>
</tr>
<tr>
<td>ROBERT</td>
<td>38,062 (131.3)</td>
<td>CHRISTINE</td>
<td>17,718 (61.1)</td>
</tr>
<tr>
<td>ANDREW</td>
<td>36,950 (127.5)</td>
<td>JEAN</td>
<td>16,919 (58.4)</td>
</tr>
<tr>
<td>CHRISTOPHER</td>
<td>34,361 (118.6)</td>
<td>HELEN</td>
<td>16,341 (56.4)</td>
</tr>
<tr>
<td>RICHARD</td>
<td>32,785 (113.1)</td>
<td>JOAN</td>
<td>16,219 (56.0)</td>
</tr>
<tr>
<td>STEPHEN</td>
<td>29,959 (103.4)</td>
<td>KATHLEEN</td>
<td>15,499 (53.5)</td>
</tr>
<tr>
<td>MARK</td>
<td>29,119 (100.5)</td>
<td>KAREN</td>
<td>15,196 (52.4)</td>
</tr>
<tr>
<td>WILLIAM</td>
<td>29,110 (100.4)</td>
<td>JANET</td>
<td>15,013 (51.8)</td>
</tr>
<tr>
<td>THOMAS</td>
<td>23,652 (81.6)</td>
<td>JULIE</td>
<td>14,940 (51.5)</td>
</tr>
<tr>
<td>ANTHONY</td>
<td>22,729 (78.4)</td>
<td>BARBARA</td>
<td>14,787 (51.0)</td>
</tr>
<tr>
<td>ALAN</td>
<td>21,567 (74.4)</td>
<td>LINDA</td>
<td>14,646 (50.5)</td>
</tr>
<tr>
<td>IAN</td>
<td>20,991 (72.4)</td>
<td>JENNIFER</td>
<td>14,536 (50.2)</td>
</tr>
<tr>
<td>BRIAN</td>
<td>18,690 (64.5)</td>
<td>DOROTHY</td>
<td>14,452 (49.9)</td>
</tr>
<tr>
<td>GEORGE</td>
<td>17,347 (59.9)</td>
<td>EMMA</td>
<td>14,404 (49.7)</td>
</tr>
<tr>
<td>DANIEL</td>
<td>17,237 (59.5)</td>
<td>JANE</td>
<td>13,792 (47.6)</td>
</tr>
</tbody>
</table>

Note: The figures in brackets are the rate per 100,000.

Source: NHSCR, (1997)
Annex F  Types of file structure

*Sequential Files or Flat Files.*

A sequential or flat file is one in which the records are considered to be stored sequentially. Access is fundamentally to one record at a time, consecutively, in the order in which they occur in the file. A search for a particular variable in each record may be possible if the file is sorted and stored on disk or in the main memory of a computer.

The basic key for records in a sequential file is the record number, identifying the sequential position of the record in the file. Other keys may exist, which can be used to determine if a given record is to be examined or passed over.

The record size determines the location of the next record for sequentially stored files. Where all the records are the same size this location is easily calculated. If the records differ in size, then either the size or the location of the next record must be a variable in each record, or each record must be fully scanned to locate its end and the beginning of the next one.

Because of their nature, sequential files are generally good for updating and listing but poor for content search, since there may be no way to access records with a specified content directly.

*Hashed files*

A hashed file is organised on the basis of the individual key values for each record in the file. From each key value, an address for the record is computed using a key-to-address transformation. Each keyword or key phrase is transformed into its numerical equivalent, from which the address can be computed. Properly done, this has the effect of scattering records uniformly throughout the computer storage. This structure enables very rapid access to each record. From the record key the system can compute the address at which the record should be stored, if it is in the file.

Unfortunately, there is always the possibility that the computations from two different key values will yield the same address, resulting in a collision. When this occurs, obviously some other location must be found for one of the records, or some blocking information needs to be added to the records that fall into the same block. Several techniques exist for doing the key-to-address transformation and to minimise the effect of any collision.

Although hashed files have excellent retrieval properties for individual records since the records are scattered throughout memory, it is very difficult to produce a sorted list of the records. In addition, it is very difficult to locate records similar to a given one. Because the assigned addresses depend heavily on the specific content of the records, similar records may be scattered widely throughout the memory. Since locating similar records is an important technique in information retrieval, hashed files find relatively little use in record linkage.

*Indexed files*

An indexed file imposes an additional structure on the records in a file, with the aim of improving access to the file. There is thus a trade-off, since creating or updating a file requires additional processing to update the index, while searching and accessing the file requires less processing. The basic purposes of an index (from the system point of view) are:

- to identify file sections of a manageable size
- to reduce the amount of work involved in a process.

The index structure divides a file into sections. Indexes may be anything that can be identified with or generated from the records. Typical indexes can be prepared that use keys based on
names, date of birth, postcode, section numbers, phonetic codes or headings, dates and assigned numbers.

The choice of an index is dictated by the logic of its use and by a balance about the file sizes. It is logical that records that will be used together should be placed within the same block. Yet, if these sections vary widely in size, then the effectiveness of indexing the file is reduced.

For very large files, multiple indexing levels are frequently desirable. In this way the size of each section can be kept relatively small, yet the sections can be organised in such a way that only a few of them need be handled at anyone time.

An inverted file is a file with an associated inverted index, which specifies those records in the file containing each of the index terms. Normally a file is only partially inverted, with only certain key terms in the inverted index. A concordance is a fully inverted file and its associated index, and specifies the exact location of each term with-in the file, right down to the page, paragraph, or sentence in the case of text files.

Tree-structured files

A tree-structured file or hierarchical file, has its records arranged in a 'tree' structure. This logical structure generally corresponds to some conceptual organisation of the data. The branching factor in a tree-structured file is the maximum number of 'children' a record can have. Similar to multilevel index structures, it is desirable to keep the branching factor reasonably low. However, because of the mirroring between the data file and the conceptual structure it is not always possible to control the branching factor.

The storage structure of a tree-structured file generally cannot correspond exactly to the tree structure, since storage has an inherently linear structure. Because the records in the file may have various sizes, particularly in a text or image file, there is no completely satisfactory way to manipulate the storage structure to mirror the logical structure of the tree.

Clustered files

The records in a clustered file are organised so that those records that have similar content using some specified criteria, are located close together and are readily accessible as a group. Because retrieval from a bibliographic or image file is inexact and relies on similarity judgements, this clustered structure is widely used in information retrieval work, and is the type of file structure most commonly encountered in record matching and linkage systems.

Development of a clustered file depends on the selection of a clustering technique, or file blocking technique. Several of these have been developed, based on statistical or set theory, or graph theory concepts. Using statistical techniques it is possible to assess, on the basis of record characteristics, the probability that a group of records which relate sufficiently to the same topic to be clustered together. Set theory techniques may be simpler. In a text file, for example, set theory clustering may be based on the number of keywords that the records have in common.

A fundamental question in forming a clustered file is whether the clusters or blocks have common members or are disjoint. In the latter case, organisation of the file is more difficult and may require multiple copies of records to be stored in the file, or multiple pointers to each record. This is the approach adopted by ORLS for the preparation of an expanded file to cope with records that contain many name variations (see Section 7.3). Maintenance of a file with overlapping clusters or blocks is difficult because of the need to coordinate the multiple copies or reference pointers each time the file is changed or updated.
Annex G Details of the selected GSS linkage projects

Case Study 1  The Comprehensive Business Directory (CBD)

Description
This is a potential application that is currently being evaluated. The aim would be to set up a Comprehensive Business Directory for government departments, as required by the Modernising Government White Paper. Currently the ONS uses matching techniques successfully to create the Inter-Departmental Business Register (IDBR). These techniques would be extended to the compilation of CBD. The compilation of CBD would require matching on data elements from different data sources and of varying data quality. The data elements to be matched would include business name, address, government reference etc. The data elements from different sources although related will not be identical. The CBD would match three categories of data:

Primary Data: e.g. PAYE information (from Inland Revenue), VAT information (from HM Customs and Excise)

Secondary Data: e.g. Information available from public registers held within government – Companies House information, OFT credit license register etc.

Direct Supply Data: Information gained from businesses directly, via electronic government services e.g. the business information website.

Matching Process
The key data for this matching application may be:

<table>
<thead>
<tr>
<th>Data sets matched and size</th>
<th>Government revenue (1.6m records)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computing environment</td>
<td>Server, IBM supplying AS400</td>
</tr>
<tr>
<td>Operating system</td>
<td>UNIX</td>
</tr>
<tr>
<td>Database for file management</td>
<td>IBM database</td>
</tr>
<tr>
<td>Matching algorithm</td>
<td>Probability</td>
</tr>
<tr>
<td>Type of matching</td>
<td></td>
</tr>
<tr>
<td>Matching variables</td>
<td>Names, address, postcode</td>
</tr>
<tr>
<td>Nature and extent of clerical matching</td>
<td></td>
</tr>
<tr>
<td>Overall match rate</td>
<td></td>
</tr>
<tr>
<td>Software employed</td>
<td></td>
</tr>
<tr>
<td>Frequency of matching</td>
<td>Quarterly</td>
</tr>
<tr>
<td>Who does matching</td>
<td>IBM/Vality</td>
</tr>
</tbody>
</table>

The exact processes to be used are yet to be determined. The CBD team plans to carry out research into the methods and software available in conjunction with other government services.

Benefits
The setting up of a CBD can deliver tangible benefits to both government and business, including:

- Promoting a more effective use of administrative data by sharing data across government departments.
• Reducing respondent burden: Ultimately we should only need to collect the same pieces of information from a business once thus significantly reducing the burden on business.

• Ensure data accuracy / verify the accuracy of information provided to us directly by businesses.

• Pull together the different information threads in order to create an overall picture.

• Ensure common standards and formats (e.g. five line standard address format) are shared and used within government.

• Seek alternative options for authenticating input other than a plethora of government Smart-cards.

Contacts

(1) Internal Paul Dunn Tel No 020 7533 5023
Office for National Statistics, 1 Drummond Gate, London SW1V 2QQ
e-mail: paul.dunn@ons.gov.uk

Case Study 2 One Number Census Matching

Description

The objective of the One Number Census (ONC) project is to estimate the level and nature of under enumeration in the 2001 Census and to integrate this with the Census data so that all Census outputs sum to One Number – the national estimate of the population on Census day. The approach is to use a post-enumeration survey to be known as the Census Coverage Survey (CCS).

A sample of postcode units is taken. Interviewers attempt to identify all households and individuals that should have responded to the Census within the sampled postcodes. Interviews will be conducted four to six weeks after Census Day. The data collected in the CCS can then be compared with the Census returns and used to estimate the total population using capture/recapture and regression methodology.

A key requirement of the ONC project is to match accurately the data collected in the CCS with those collected in the Census. Since the records of interest in the matched dataset are those that have not been linked, the accuracy of the matching is of critical importance. In the 1991 Census, the overall level of underenumeration was estimated at two per cent. Therefore, missing just one per cent of true matches could lead to an inflation of almost 50 per cent in the records of interest.

Matching Process

The key data for this matching application is expected to be:

<table>
<thead>
<tr>
<th>Data sets matched and size</th>
<th>2001 Census (55m), Census Coverage Survey (0.5m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computing environment</td>
<td>Server and Local Intranet</td>
</tr>
<tr>
<td>Operating system</td>
<td>Windows and Unix</td>
</tr>
<tr>
<td>Database for file management</td>
<td>SYBASE</td>
</tr>
<tr>
<td>Matching algorithm</td>
<td>Exact or Probabilistic</td>
</tr>
<tr>
<td>Type of matching</td>
<td>Across data files</td>
</tr>
<tr>
<td>Matching variables</td>
<td>Key demographic variables such as name, address and date of birth</td>
</tr>
<tr>
<td>Nature and extent of clerical matching</td>
<td>Clerical intervention if the matching criteria below the pre-set threshold</td>
</tr>
<tr>
<td>Overall match rate</td>
<td>Unknown at this stage</td>
</tr>
<tr>
<td>Software employed</td>
<td>Bespoke in-house</td>
</tr>
<tr>
<td>Frequency of matching</td>
<td>Once every ten years</td>
</tr>
<tr>
<td>Who does matching</td>
<td>In-house (ONS Census Division)</td>
</tr>
</tbody>
</table>

The proposed strategy is to match records within postcodes using a combination of probabilistic and clerical matching. Since it is possible that the CCS interviewers may stray outside the postcode of interest, postcodes contiguous to the sampled postcodes will also be searched.

The clerical part of the matching process will resolve all the undetermined cases for which the matching criteria falls below a pre-set threshold. An analysis of the Dress Rehearsal Data and the following test CCS will include an assessment of the utility of names and addresses as matching variables, determine match weights, set up thresholds and determine the overall level of matching achieved.

**Benefits**

Census data is used to allocate large amounts of government spending. It is therefore important for the data to be as accurate as possible. The 1991 Census is thought to have had disproportionately high levels of under-enumeration amongst young men, babies and the elderly. An accurate assessment of the undercount directly affects the accuracy of resource allocation programmes.

The benefit of the ONS project, and therefore the matching, is the improvement in the quality of the Census data and results.

**Contacts**

(1) Internal: Jennet Woolford  
Office for National Statistics, Segensworth Road, Titchfield Hants PO15 5RR  
e-mail: jennet.woolford@ons.gov.uk  
Tel no: 01329 813947

**Case Study 3  Retail Planning and GIS**

**Description**

The purpose of matching is to define statistical areas of town centre activity and generate statistics for the areas. The data from a number of sources such as employment and turnover data from the Inter-Departmental Business Register (ONS) and floorspace data from the Valuation Office Agency is linked together to generate town centre areas and compile statistics. Currently, matching is undertaken at the aggregated level of unit postcode. In future, it is planned to match individual businesses, properties and locations using National Land and Property Gazetteer.
**Matching Process**

The key data for this matching application is:

| Data sets matched and size | ONS IDBR Employment (0.7m records for London and South West, 26.7MB), ONS IDBR Turnover (0.7m records for London and South West, 26.7MB), VOA Floor Space (0.5m records for London and South West, 16MB), OS Code-Point (1.6 m, 103MB), CPD-1998Q3 (2.0m, 173MB) |
| Computing environment     | Server and Local Intranet |
| Operating system          | Windows and Unix |
| Database for file management | ORACLE |
| Matching algorithm        | Exact Matching |
| Type of matching          | Set link between different files |
| Matching variables        | Organisation Identifiers (Postcode) |
| Nature and extent of clerical matching | Clerical matching employed |
| Overall match rate        | 98%-100% |
| Software employed         | Proprietary |
| Frequency of matching     | Annual |
| Who does matching         | University College London |

The data cleaning part of the matching process is in two stages: some cleaning done on source files (IDBR, VOA, OS Code Point and CPD) and some later jointly on ORACLE. The clerical part of the matching process involves making IDBR employment data available to local authorities on the internet and those in turn verifying the data.

There are some problems in the matching methodology in that some business addresses are PO Box referenced to a postcode centroid thereby in some cases seriously misplacing the business. In the future, when matching individual business there are expected to be problems with the quality of the ‘address’ data.

**Benefits**

Matching administrative records on businesses and properties has been essential for defining town centre activity and improving statistics for retail planning.

**Contacts**

(1)Internal: Stephen Hall  
Department of the Environment, Transport and the Regions, Eland House, Bressenden Place, London SW1E 5DU  
e-mail: Stephen_Hall@detr.gsi.gov.uk

(2) External: UCL London

**Case Study 4  Secondary School performance tables and Key Stage 3 to GCSE value added**

**Description**

The purpose of the matching is to create a single data base of qualifications by linking together Key Stage 3 results, GCSE results and GNVQ results for each pupil. These results come from a variety of sources, and at different time periods. The database is used
extensively for the production of annual school and college performance tables, monitoring progress towards the National Targets for Education and Training (for the 'foundation targets' relating to young people), and for a range of other analyses including calculation of 'value added' for schools.

There are available for each pupil 3 separate KS3 results (English, Maths and Science) all coming from a single source. The KS3 results are available from 2 years earlier than GCSE results and arrive as 3 separate files. The GCSE records are available as entries for the summer session in June each year, and as results in August and in the late spring (winter results). There are 3 GCSE awarding bodies and separate files are obtained from each of these. The GNVQ records are available 3 times a year (November, May and August), with registration data available for up to two years before the results data become available in August of the year in which pupils become 15. The linkage of these results creates a single record containing cumulative performance at KS 3, GCSE and foundation and intermediate GNVQ.

**Matching Process**

The key data for this matching application is:

<table>
<thead>
<tr>
<th>Data sets matched and size</th>
<th>Key Stage 3 results, GCSE results, GNVQ results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computing environment</td>
<td>Server and Local Intranet</td>
</tr>
<tr>
<td>Operating system</td>
<td>Windows and Unix</td>
</tr>
<tr>
<td>Database for file management</td>
<td>ORACLE</td>
</tr>
<tr>
<td>Matching algorithm</td>
<td>Exact</td>
</tr>
<tr>
<td>Type of matching</td>
<td>Across data files</td>
</tr>
<tr>
<td>Matching variables</td>
<td>Name, Date of birth, Institution,</td>
</tr>
<tr>
<td>Nature and extent of clerical matching</td>
<td>Matched record are supplied to schools for manual checking.</td>
</tr>
<tr>
<td>Overall match rate</td>
<td></td>
</tr>
<tr>
<td>Software employed</td>
<td>Proprietary software</td>
</tr>
<tr>
<td>Frequency of matching</td>
<td>Annual</td>
</tr>
<tr>
<td>Who does matching</td>
<td>Univ. Bath (SERAP)</td>
</tr>
</tbody>
</table>

The process of matching involves matching the three KS3 results (English, Maths and Science) into one and then matching onto GCSE entries and GNVQ registrations followed by merging/matching with the GCSE and GNVQ results as and when they become available.

The clerical intervention part of the matching process involves following known data problems and checking a sample of matched records to test for unexpected systematic or other errors and to generally confirm the quality of the process. The success of the matching process is measured in terms of corrections returned from schools estimated to be under 1 per cent.

**Benefits**

The main benefit of the matching is to reduce burden on schools of the performance tables, by providing them with a pre-matched set of results which they can check.

**Contacts**
(1) Internal: Arthur Barnett
Department for Education and Employment, Caxton House, Tothill Street,
London SW1H 9NF

Tel no: 020 7273 5935
Case Study 5  Cross matching of the NHSCR

Description
The objective of this study is to match the NHSCR against itself to estimate the number of duplicate entries on the register. The register was built from the 108 files prepared by each of the FHSAs (now DHAs) and merged together. Some of these people will have been included on two or more of the individual registers, under slightly different identifiers. People who have left the UK for short periods and subsequently returned may have been issued with a new NHS number.

A complete sample of the NHSCR was extracted and prepared for internal cross matching. The parallel files containing the changes in names, date of birth and address were also extracted. The files were merged and expanded as described elsewhere in this report (Section 6), the 57 million records upon expansion and merging resulting in 81 million records.

Matching Process
The key data for this matching application is:

| Data sets matched and size          | NHSCR (55 million, expanded to 81 million) |
| Computing environment              | Server and Local Intranet                  |
| Operating system                   | Windows 98 Unix                           |
| Database for file management       | Flat file                                  |
| Matching algorithm                 | Exact or Probabilistic                     |
| Type of matching                   | Prepare a file of duplicates for clerical check and subsequent marking/deletion |
| Matching variables                 | Names, gender, date of birth and cipher    |
| Nature and extent of clerical matching | Clerical intervention if the matching criteria below the pre-set threshold |
| Overall match rate                 | 93–95%                                     |
| Software employed                  | Bespoke in-house prepared by ORLS          |
| Frequency of matching               | Once only                                  |
| Who does matching                   | OMLS for NHSCR                             |

The proposed strategy is to match records within surname or the phonetic code of the surname and then use the other matching variables. The other matching orders included first forename (the NHSCR only contains 1 forename and up to 4 initials) and date of birth. All the three matches were merged together and computerised files prepared for printing and clerical checking.

Benefits
The NHSCR is a register of persons who are registered with the NHS and is kept up to date by returns from the vital registration service and from GPs through the District Health Authority. Transfers of patients between Districts in England and Wales (Scotland and Northern Ireland have their own registers) are recorded in the register and from this
information it is possible to monitor the migration and for making corrections to the mid-year population estimates.

The NHSCR is used for a variety of research purposes, particularly medical, and for flagging cohorts of people who are in research studies.

Contacts
(1) Internal: Chris Cotton Tel no: 01329 813947
NHSCR, Trafalgar Road, Southport, Lancashire,

(2) External: Leicester Gill Tel no: 01865 227017
University of Oxford, Unit of Health-Care Epidemiology,
Institute of Health Sciences, Old Road, Headington, Oxford, OX3 7LF

e-mail: leicester.gill@dphpc.ox.ac.uk

Case Study 6  Building a linked file of hospital and vital records

Description
The objective of this study is to match together hospital discharge records and death registration details to prepare a file suitable for a number of analytical procedures. Although the names and postcode are used as part of the matching process, they are stripped off the analytical file and never used in the analyses.

Matching Process

The key data for this matching application is expected to be:

<table>
<thead>
<tr>
<th>Data sets matched and size</th>
<th>Hospital and Death records spanning 38 years</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computing environment</td>
<td>Local Intranet</td>
</tr>
<tr>
<td>Operating systems</td>
<td>Windows 98, Unix</td>
</tr>
<tr>
<td>Database for file management</td>
<td>Flat file</td>
</tr>
<tr>
<td>Matching algorithm</td>
<td>Exact and Probabilistic</td>
</tr>
<tr>
<td>Type of matching</td>
<td>Prepare a file of containing the records for every person on the file linked together and stored in temporal order.</td>
</tr>
<tr>
<td>Matching variables</td>
<td>Names, gender, date of birth and address, postcode and other variables</td>
</tr>
<tr>
<td>Nature and extent of clerical matching</td>
<td>Clerical intervention if the matching criteria below the pre-set threshold</td>
</tr>
<tr>
<td>Overall match rate</td>
<td>98–99%</td>
</tr>
<tr>
<td>Software employed</td>
<td>Bespoke in-house prepared by ORLS</td>
</tr>
<tr>
<td>Frequency of matching</td>
<td>Add the data for each year and periodically cross match the whole file against itself</td>
</tr>
<tr>
<td>Who does matching</td>
<td>in-house: ORLS</td>
</tr>
</tbody>
</table>

The proposed strategy is to match records within surname or the phonetic code of the surname and then use the other matching variables. The other matching orders include first forename, date of birth and postcode. Where the NHS number is available, this is used firstly in a exact match step and secondly as part of the probabilistic match. All the matches are merged together and computerised files prepared for printing and clerical checking.
Benefits
See Section 2 of the report

Contacts
(1) External: Leicester Gill  Tel no: 01865 227017
University of Oxford, Unit of Health-Care Epidemiology, Institute of Health Sciences, Old Road, Headington, Oxford, OX3 7LF
e-mail: leicester.gill@dphc.ox.ac.uk
Glossary and abbreviations

Glossary

ASCII Standard codes to represent characters in one byte, defined by the American National Standards Institute.

Ad hoc request A request for information that has not been previously specified.

Ad hoc retrieval standard retrieval task in which the user specifies his information need through a query that initiates a search (executed by the information system) for records which are likely to be relevant to the user.

Algorithm a description of an ordered and logical series of steps to be followed to solve a problem. This is usually in the form of a computer program.

Bias (systematic) Any trend in the collection, analysis and interpretation of data that can lead to conclusions that are systematically different from the truth.

Binit Weight The outcome specific weight computed for the agreement between two variables and represented in the form of logarithms to the base 2.

Blocking the use of sequencing information to divide files into blocks. This method reduces the search for a match to those records in the same block.

Boolean logical operator A group of search terms (e.g. 'AND', 'OR', 'NOT') available on the most common and widely used search engines, which help in refining the search strategy and retrieving information from a computer database.

Boolean model a classic model of record retrieval based on classic set theory.

Browsing interactive task in which the user is more interested in exploring the document collection than in retrieving documents which satisfy a specific information need.

Byte a computer representation of a character (q.v.).

Character a unit of alphanumeric data. One of a set of symbols used to denote an alphabetic letter (a-z, A-Z), a number (0-9), or a special character (+=@*, etc).

Check characters A letter or number incorporated into a persons NHS or other identity number, to provide a check on the validity of the number.

CHI number Community Health Index number issued in Scotland.

Clustering the grouping of records which satisfy a set of common properties. The aim is to assemble together records which are related among themselves.

Coding the substitution of text symbols by numeric codes with the aim of encrypting or compressing text.
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compression of text</td>
<td>the study of techniques for representing text in fewer bytes or bits.</td>
</tr>
<tr>
<td>Concatenation</td>
<td>the concatenation of two strings $a$ and $b$, $ab$ is the string obtained by appending $b$ to $a$.</td>
</tr>
<tr>
<td>Content-based query</td>
<td>query exploiting data content.</td>
</tr>
<tr>
<td>Conversion</td>
<td>changing from one form to another, as in converting from analogue to digital.</td>
</tr>
<tr>
<td>Cumulative file</td>
<td>a file that file that contains records for a cohort of people collected at different time points.</td>
</tr>
<tr>
<td>Data</td>
<td>raw facts that have been collected, organised and stored in a computer file.</td>
</tr>
<tr>
<td>Data file</td>
<td>This data file will be matched against the master file to establish links between the records.</td>
</tr>
<tr>
<td>Data mining</td>
<td>extraction of new data, relations, or partial information from any type of data file.</td>
</tr>
<tr>
<td>Data retrieval</td>
<td>the retrieval of variables (tuples, objects, Web pages, documents) whose contents satisfy the conditions specified in a user query.</td>
</tr>
<tr>
<td>Deterministic match</td>
<td>see Exact match.</td>
</tr>
<tr>
<td>Document</td>
<td>a unit of retrieval. It might be a paragraph, a section, a chapter, a Web page, an article, or a whole book.</td>
</tr>
<tr>
<td>Edit distance</td>
<td>(between two strings) minimum number of insertions, deletions and replacements of characters necessary to make two strings equal.</td>
</tr>
<tr>
<td>Exact match</td>
<td>mechanism by which only the objects exactly satisfying some well specified criteria are matched together.</td>
</tr>
<tr>
<td>Expert system</td>
<td>a computer system that uses knowledge and inference procedures to solve problems that would require significant human expertise for their solution.</td>
</tr>
<tr>
<td>Field</td>
<td>In this context, a field is a piece of identifying information, for example, a name, date of birth etc. The variable is sometimes called a variable or an item (q.v.).</td>
</tr>
<tr>
<td>File</td>
<td>a collection of similar records.</td>
</tr>
<tr>
<td>Flat file</td>
<td>a file containing only fixed length records sorted into some arbitrary order.</td>
</tr>
<tr>
<td>Frequency ratio</td>
<td>The frequency of a given comparison outcome amongst correctly linked pairs, divided by the corresponding frequency among the unlinked pairs selected from a random sample.</td>
</tr>
<tr>
<td>Full text</td>
<td>a logical view of the documents in which all the words which compose the text of the document are used as indexing terms.</td>
</tr>
<tr>
<td>Fuzzy</td>
<td>a method of reasoning that resembles human reasoning since it permits approximate values and inferences and incomplete and ambiguous data.</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Fuzzy data</td>
<td>data that is incomplete or ambiguous.</td>
</tr>
<tr>
<td>Fuzzy model</td>
<td>a set theoretic model of document retrieval based on fuzzy theory.</td>
</tr>
<tr>
<td>Hashing</td>
<td>a method of accessing a file in which the address or key is calculated from the key data variable.</td>
</tr>
<tr>
<td>Heuristic methods</td>
<td>proceeding to get a solution by trial and error methods.</td>
</tr>
<tr>
<td>Huffman coding</td>
<td>an algorithm for coding text in which the most frequent symbols are represented by the shortest codes.</td>
</tr>
<tr>
<td>IF/THEN/ELSE</td>
<td>a programming construct in which one of two possible outcomes is taken, depending on whether the logical condition is TRUE or FALSE.</td>
</tr>
<tr>
<td>Index</td>
<td>a mechanism by which a program orders the records in a file.</td>
</tr>
<tr>
<td>Index point</td>
<td>the initial position of a text element which can be searched for, for example a word.</td>
</tr>
<tr>
<td>Index term</td>
<td>(or keyword) a pre-selected term which can be used to refer to the content of a document. Usually, index terms are nouns or noun groups. In the Web, however, some search engines use all the words in a document as index terms.</td>
</tr>
<tr>
<td>Information retrieval</td>
<td>(IR) part of computer science which studies the retrieval of information (not data) from a collection of written documents. The retrieved documents aim at satisfying a user information need usually expressed in natural language.</td>
</tr>
<tr>
<td>Intranet</td>
<td>an Internet type network built inside an organisation, which may or may not be connected to the Internet itself.</td>
</tr>
<tr>
<td>Item</td>
<td>In this context, an item is a piece of identifying information, for example, a name, date of birth etc. The variable is sometimes called a field or an variable (q.v.).</td>
</tr>
<tr>
<td>Keyword</td>
<td>see Index term.</td>
</tr>
<tr>
<td>Levenstein distance</td>
<td>a distance measure between two strings, given by the maximum number of symbol insertions, deletions and substitutions required to transform one string into another string.</td>
</tr>
<tr>
<td>Lexicographical order</td>
<td>order in which the words are listed in a dictionary or telephone directory.</td>
</tr>
<tr>
<td>Lexicon</td>
<td>a look-up table for conversion of one string into another string.</td>
</tr>
<tr>
<td>Linkage</td>
<td>In the broadest sense, Record Linkage is the bringing together of information from two or more records that are believed to belong to the same person or entity.</td>
</tr>
<tr>
<td>Log2</td>
<td>are logarithms to the base 2 and are calculated using logarithms to the base 10 in the following fashion</td>
</tr>
<tr>
<td></td>
<td>$\log_2 W = \frac{\log_{10} W}{\log_{10} 2} = \frac{\log_{10} W}{0.30103}$</td>
</tr>
</tbody>
</table>
Longitudinal a study that looks at the flow or sequence of events occurring over a given period of time to particular groups or cohorts of people. (see cumulative file).

Master File a file that is used as the primary source of data for a given job, it is relatively permanent, even though its contents may change.

Metasearch a search technique common on the World Wide Web where a single point of entry is provided to multiple search engines. A meta search system sends a user's query to the back-end search engines, combines the results, and returns a single, unified hit-list to the user.

Metadata attributes of data or a record, usually descriptive as author or content, often broken up into categories or facets, and typically maintained in a catalogue.

Modulus a number used as a divisor in check digitizing systems.

Natural language the ordinary spoken language that humans use in everyday conversation.

Nearest-neighbour a query that requests the spatial object closest to the specified object. Query.

Near-neighbour Words that are in close proximity to each other, for example BETSY and BETTY.

NHS Number National Health Service Number is issued to everyone who is registered with a General Practitioner. The Number is issued for England and Wales by the NHS Central Register (NHSCR) in Southport, for Scotland by NHSCR in Edinburgh, and for Northern Ireland by NHSCR in Belfast.

Null Hypothesis Hypothesis put forward when carrying out statistical significance tests, which states that: a) there are no differences between groups being compared, or b) there is no association or relationship between variables, in the studied population.

ORLS The Oxford Record Linkage System is a collection of person linked hospital discharge abstracts and vital records spanning 38 years.

Parsing resolve a sentence or string into its component parts and describe the various parts grammatically.

Phoneme any of the units of sound in a specified language that distinguishes one word from another (see phonetic).

Phonetic deals with the main sound units of speech and provides a direct correspondence between symbols and sound (see phoneme).

Population Population for which the results of a given investigation are to be analysed. The distinction needs to be drawn between the wider population and the sample population, this being a subset of the former.

Probabilistic model a classic model of record retrieval based on a probabilistic interpretation of record relevance (to a given user query).
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query or data record</td>
<td>The record which is to be matched against the master file.</td>
</tr>
<tr>
<td>Query language</td>
<td>a computer language used to retrieve specific information from a data base.</td>
</tr>
<tr>
<td>Record</td>
<td>a unit of information that contains a group of related data variables.</td>
</tr>
<tr>
<td>Rule-based system</td>
<td>a form of expert system in which human knowledge is captured as a series of IF/THEN/ELSE rules concerning objects or events.</td>
</tr>
<tr>
<td>Semantics</td>
<td>the meaning of words and phrases.</td>
</tr>
<tr>
<td>Sort key</td>
<td>a key used as the basis for reorganising the sequence of variables in a file or dataset.</td>
</tr>
<tr>
<td>Stemming</td>
<td>a technique for reducing words to their grammatical roots.</td>
</tr>
<tr>
<td>Syntax</td>
<td>rules that are applied to matching words in order to form sentences.</td>
</tr>
<tr>
<td>Text file</td>
<td>a computer file that contains words and characters. Such files are commonly created in word processing activities.</td>
</tr>
<tr>
<td>Text structure</td>
<td>information present in a text apart from its content, which relates its different portions in a semantically meaningful way.</td>
</tr>
<tr>
<td>Thesaurus</td>
<td>a data structure composed of (1) a pre-compiled list of important words in a given domain of knowledge and (2) for each word in this list, a list of related (synonym) words.</td>
</tr>
<tr>
<td>Type I error</td>
<td>In the context of record matching, a Type I error is where records that refer to the same person have failed to match together.</td>
</tr>
<tr>
<td>Type II error</td>
<td>In the context of record matching, a Type II error is where records have been matched together that in fact belong to two or more different people.</td>
</tr>
<tr>
<td>Variable</td>
<td>In this context, a variable is a piece of identifying information, for example, a name, date of birth etc. The variable is sometimes called a field or an item (q.v.).</td>
</tr>
<tr>
<td>Wild-card character</td>
<td>a character that will match any character or sequence of characters in a name or string. Typical wildcard characters are &quot;*&quot; and &quot;?&quot;</td>
</tr>
<tr>
<td>Abbreviations</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>ANSI</td>
<td>American National Standards Institute.</td>
</tr>
<tr>
<td>ASCII</td>
<td>American Standard Code for Information Interchange.</td>
</tr>
<tr>
<td>CHI</td>
<td>Community Health Index.</td>
</tr>
<tr>
<td>DETR</td>
<td>Department of Environment, Transport and the Regions.</td>
</tr>
<tr>
<td>DfEE</td>
<td>Department for Education and Employment.</td>
</tr>
<tr>
<td>DH</td>
<td>Department of Health.</td>
</tr>
<tr>
<td>DHA</td>
<td>District Health Authority.</td>
</tr>
<tr>
<td>EPR</td>
<td>Electronic Patient Record.</td>
</tr>
<tr>
<td>FORTRAN</td>
<td>FORmula TRANslator, a high level computing language.</td>
</tr>
<tr>
<td>GP</td>
<td>General Practitioner (Family Doctor).</td>
</tr>
<tr>
<td>GSS</td>
<td>Government Statistical Service.</td>
</tr>
<tr>
<td>HES</td>
<td>Hospital Episode Statistics.</td>
</tr>
<tr>
<td>KMP</td>
<td>Knuth, Morris, Pratt algorithm.</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network.</td>
</tr>
<tr>
<td>NHS</td>
<td>National Health Service.</td>
</tr>
<tr>
<td>NHS number</td>
<td>National Health Service number issued by the NHSCR (qv).</td>
</tr>
<tr>
<td>NHSCR</td>
<td>National Health Service Central Register.</td>
</tr>
<tr>
<td>NSTS</td>
<td>National Strategic Tracing Service.</td>
</tr>
<tr>
<td>NYSIIS</td>
<td>New York State Identification and Intelligence System code.</td>
</tr>
<tr>
<td>ONCA</td>
<td>Oxford Name Compression Algorithm.</td>
</tr>
<tr>
<td>ONS</td>
<td>Office for National Statistics.</td>
</tr>
<tr>
<td>ORLS</td>
<td>Oxford Record Linkage Study.</td>
</tr>
<tr>
<td>RAM</td>
<td>Random Access Memory.</td>
</tr>
<tr>
<td>SES</td>
<td>Socio-Economic Status.</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language.</td>
</tr>
<tr>
<td>WWW</td>
<td>World Wide Web.</td>
</tr>
</tbody>
</table>
References and bibliography

For completeness other important references not quoted in the text are also included.


Brown D A H. Biquinary decimal error correction codes with one, two and three check digits. The Computer Journal 17 3 (1973), 201–204.


Roos L L, Roos N P, Cageorge S M and Nicol J P. How good are the data? Reliability of one


Scheuren F and Winkler W E. Regression analysis of data files that are computer matched, II. *Survey Methodology* 23 (1997), 157–165.


<table>
<thead>
<tr>
<th>Report</th>
<th>Title</th>
<th>Publisher</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report 1</td>
<td>Dave Elliot. <em>Software to weight and gross survey data with applications to the EC Household Panel and Family Expenditure Surveys,</em> (Government Statistical Service: 1997)</td>
<td></td>
</tr>
<tr>
<td>Report 6</td>
<td>Michael A Baxter. <em>Interpolating annual data into monthly or quarterly data,</em> (Government Statistical Service: 1998)</td>
<td></td>
</tr>
</tbody>
</table>
Publications in the National Statistics Methodology Series continued


Available only from the Office for National Statistics – see page 2 for contact points. Price £5 unless otherwise stated. Copies are available free of charge to GSS members.