National Records of Scotland

Administrative Data Based Population Estimates Scotland 2016: Methodology Report

Statistical Research

Published on 17 November 2020

Disclaimer: These statistical research outputs are **not the OFFICIAL STATISTICS** for Population Estimates for Scotland. The Official Statistics can be found at the statistics and data section of National Records of Scotland's website.

This publication reports on the results of research into how population estimates might be produced using a range of administrative data.

Any presentation or use of these research outputs should make clear to users the nature and purpose of the statistics.

Contents

1. Abstract

National Statistics for population estimates in Scotland are provided by the census and the mid-year population estimates (both published by National Records of Scotland (NRS)). NRS is now exploring the possibility of using de-identified data from public-sector administrative datasets to produce population estimates.

The datasets are first separated into two dataset: one with variables used for linking (e.g. name and date of birth), and the other with analysis variables (e.g. age). The variables for linking are standardised into: first, middle and last names, day, month and year of birth, gender (which may be derived from a variable recording gender or from a variable recording sex) and postcode. From these a range of linking variables are derived, such as nicknames and parts of hyphenated names. These are then scrambled in such a way that the same value always gives the same scrambled text, but that the original text cannot be recovered from the scrambled text, making the dataset more secure.

The analysis variables and the scrambled linking variables are sent to a secure facility. The analysts working at that location were different from those who had handled the original data (for security purposes). Here, linking variables are merged back on to the analysis variables for each data source. The different data sources are then linked together using the scrambled linking variables. Some links will have exact agreement on the standardised variables (first name, last name, postcode and date of birth). For others some information may not be available, so a subset of these must be used (for example, electoral register data does not include date of birth, the scrambled name and postcode are used). In other cases the information may be recorded differently, and so the derived linking variables, such as nicknames or name parts, are used. A score is assigned to each link, based on how well the two records agree. The score is also amended if the combination of variable values are not rare. For example it is more likely that there are two different John Smiths with the same date of birth, than two Sarah-Jane Watt-Maxwells.

Using the links found, and the scores for each of them, the records from the different sources are grouped together in such a way that each group is expected to represent a unique person. As some datasets contain records for people who are no longer at that location, the groups are trimmed down to only those groups where the person is expected to be at that location. Groups where there is a record on the death registrations for the deceased where the date of death is before 30/6/2016 are excluded. Groups where the NHSCR (National Health Service Central Register) indicates that the person has died or left the country are also excluded. The remaining groups will be included if they meet one of the following conditions:

- appears on the NHSCR and appears on one of the other datasets;
- appears on the birth registrations as a child and is aged below one.

For each group an age, sex and (scrambled) postcode is identified. Where there is conflict between different datasets on these then the values from NHSCR are prioritised. From the scrambled postcode a lookup table is used to assign it to a council area, SIMD (Scottish Index of Multiple Deprivation) decile and urban–rural category. An age–sex breakdown by council area, SIMD or urban–rural category can then be found.

2. Method Summary

The following is an outline of the steps involved in the new method. These steps are discussed in detail in Section 4.

1. Standardise the identifiable data
2. Separate payload and identifiable data
3. Generate linking variables
    a. Name variables
    b. Postcode levels
    c. Date of birth Bloom filters
4. De-identify
5. Transfer data to the National Safe Haven
6. Join payload and de-identified datasets
7. Concatenate Datasets and Link
8. Trim Links
9. Resolve Linked Records to Individuals
10. Apply Business Rules
11. Assign Analysis Variables

Linking multiple datasets together could potentially increase their sensitivity. Therefore, the datasets to be linked are only brought together in the same processing environment once they have been de-identified. In order to maintain high levels of security, and preservation of privacy, the process is divided into three parts. Each part processed in different environments, and by different individuals. In this way, none of the team (apart from the NRS Head of Admin Data) see the identifiable data of multiple datasets simultaneously (see Figure 1).

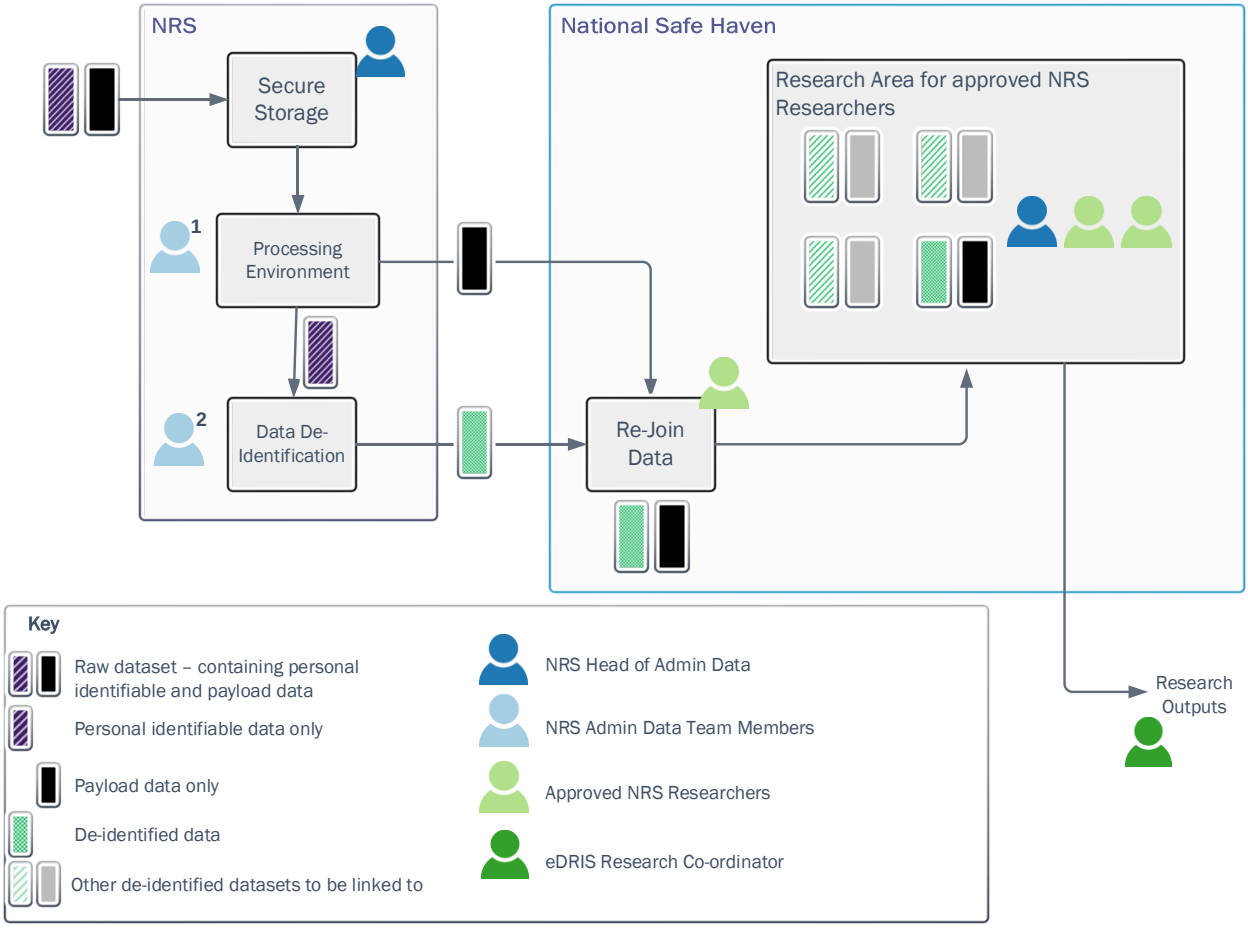

Figure 1 Separation of functions among team members.

The three parts are:

1. NRS: Processing Environment. Here the identifiable data is standardised and the payload data (for example, age, sex, NHSCR posting) is separated and sent directly to the National Safe Haven.
2. NRS: De-identification. Here the identifiable data is de-identified so that the individuals cannot be identified. The de-identified dataset is sent to the National Safe Haven.
3. National Safe Haven. Here the payload and de-identified data are merged back together. The datasets from the different sources are then linked and processed to produce Scotland's Integrated Demographic Dataset (SIDD).

More-detailed information on the separation of functions and security is included in the Data Protection Impact Assessment (DPIA). The flow through the methodology steps is summarised in the flow chart in Figure 2.

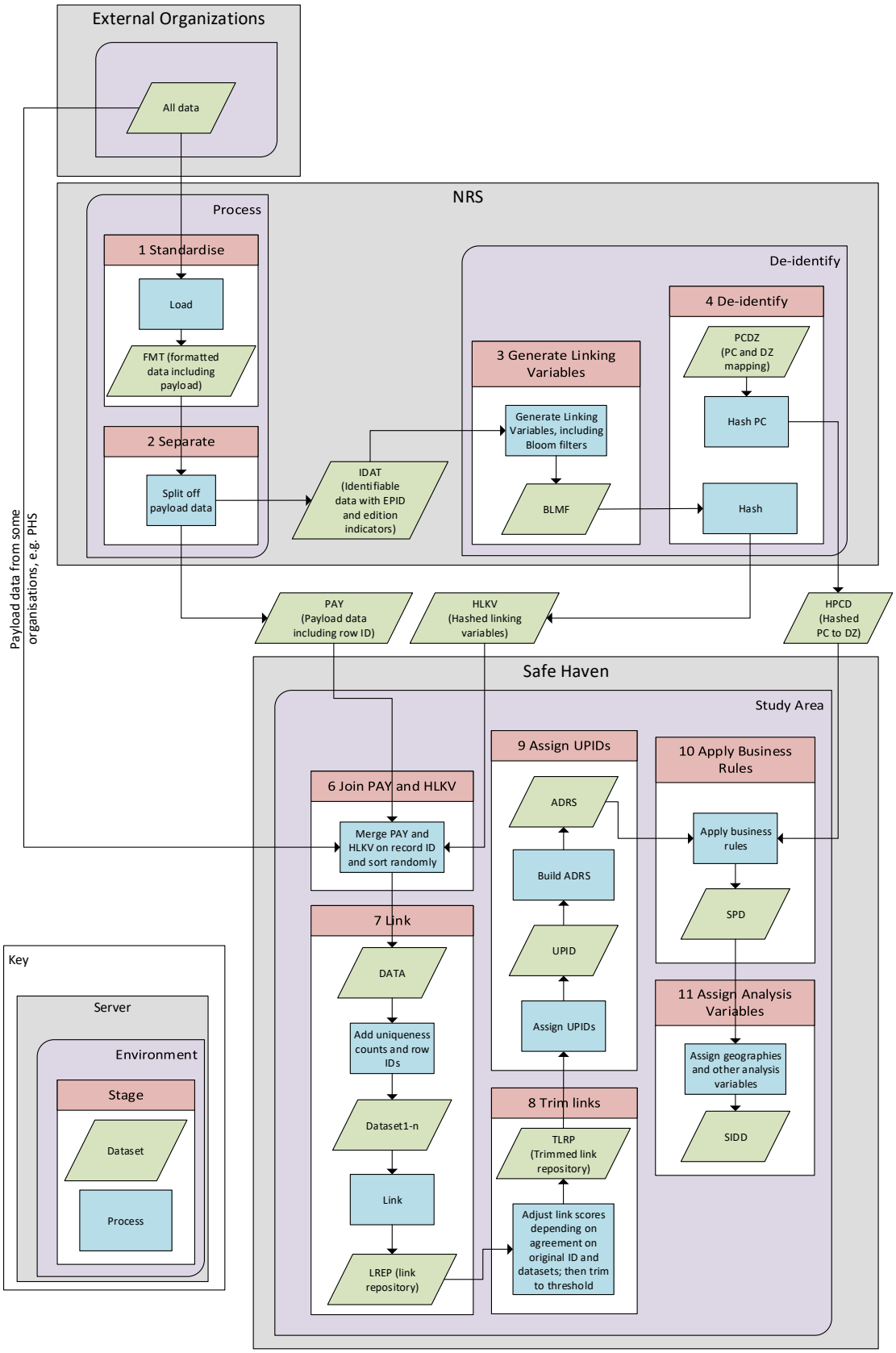

Figure 2 Flow of data (green parallelograms) and steps (blue rectangles) through the process. The numbers for each stage correspond to the stages described in the text.

3. Datasets Used

A range of administrative datasets are provided by suppliers. These are given in Table 1.

Table 1 Datasets used along with the supplier that provided them and a brief description.

Name	Source[1]	Description
NHSCR	NRS	People born or died in Scotland or registered with a Scottish GP.
Heath Activity	PHS	Health activity data (including secondary care, prescriptions, dental and screening).
Electoral Register	EROs	People registered to vote, including for Scottish elections (16–17 year olds). Dates of birth are not supplied.
School Pupil Census	SGLD	Children and young people enrolled at publically funded schools in Scotland. Names not supplied.
Vital Events	NRS	Births, deaths and marriages registered in Scotland, including the details of the parents of children born and those registering deaths.
HESA	HESA	All students studying at Scottish higher education providers and Scottish domiciled students studying at higher education providers in England, Wales and Northern Ireland. One HESA dataset covered 2015/16, and another 2016/17.
FES	SFC	Students studying in colleges in Scotland, except those studying in Higher Education programmes in the University of Highlands and Islands or Scotland's Rural College.
ROS	ROS	Data relating to the sale of properties in Scotland.
Geography	NRS	Lookups from postcode to council area, SIMD and urban–rural classifications.

The National Health Service Central Register (NHSCR) contains basic demographic details of everyone who was born, or have died, in Scotland plus anyone else who is (or has been) on the list of a general medical practitioner (GP) in Scotland. The Register exists mainly to allow the smooth transfer of patients who move between Health Board areas (or across borders within the UK).

The NHSCR is therefore expected to include almost all the people in Scotland. In addition, it includes people who were previously registered with a GP in Scotland, but have since died or registered with an NHS GP elsewhere in the UK. This is particularly useful as it allows identification of individuals who are no longer in the Scotland population (but who may appear on one of the other administrative datasets used). These strengths of the NHSCR for population estimates, raise the

[1] See Glossary in Annex 1 for expansion of abbreviations.

possibility of using it in isolation for population estimates. To explore this possibility the NHSCR is explored below.

The status of people recorded on the NHSCR is indicated using a variable called the posting. Figure 3 shows the age distribution of the people on the NHSCR, broken down by groupings of postings, compared with the published 2016 Mid-Year Population Estimates for Scotland (MYE).

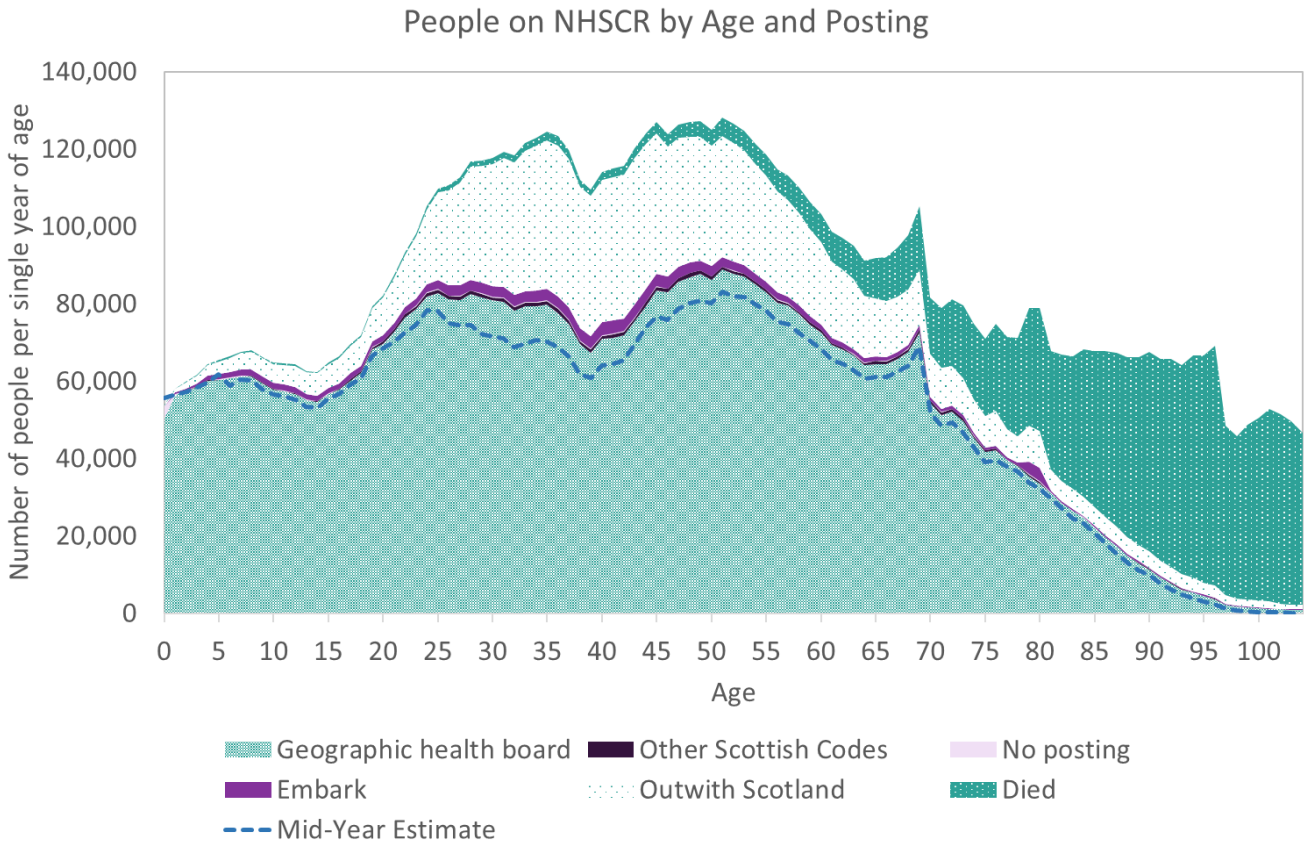

Figure 3 Number of people on the NHSCR by age and posting, compared with the 2016 Mid-Year Population Estimates for Scotland. Embark relates to where the person is believed to have left the UK. Outwith Scotland relates to people who have moved elsewhere in the UK (and registered with a GP).

Figure 3 demonstrates that, as expected, there are many more people on the NHSCR than there are people in the population. Therefore, many of these people will need to be excluded from the estimates. The postings variable is a useful way to do this. The first group that should be excluded are people who have died. It can be seen in Figure 3 that the NHSCR retains records for people who have died. Removing these records brings the count much closer to the MYE for older people.

There is another large group who could be excluded to improve estimates. These are people who have been registered with a GP in Scotland at some time, but have

since registered with a GP elsewhere in the UK. On registering with a GP, the patient's new GP will request their medical records from their previous GP. This informs their previous GP that the patient should be deregistered with them. These patients will remain on the NHSCR, but will be flagged to indicate the NHS trust where they are now registered. (This is particularly useful as it identifies people who may appear in other administrative datasets, allowing them to be excluded.) Removing this group brings the counts much closer to the MYE, especially for people aged around 20–65. Similarly, excluding the embark group (who are believed to have left the UK completely) also improves the estimate for most age groups.

It can be seen, however, that although excluding these three groups greatly improves the estimate, it remains higher than the MYE. (When the administrative data based population estimates are higher than the MYE this is hereafter referred to as over-coverage, and if it is lower then this is referred to as under-coverage. The MYE will be used as a benchmark for the purposes of this work.) One main reason for this would be people who leave the UK and do not deregister with their GP. The over-coverage appears for all ages apart from zero year olds, and to a lesser extent five year olds. Note that it can be up to 21 days between a baby being born and their birth being registered. Babies will appear on the NHSCR once their birth has been registered. Therefore, on any given day there will be babies in the population who do not appear on the NHSCR. The dataset available to the project is of all the people on the NHSCR on the 30th of June 2016, and so there will likely be babies who have been born in the latter part of June who will appear on NHSCR after the 30th of June.

In order to achieve population estimates that are more accurate than can be achieved by the NHSCR dataset on its own, the NHSCR is linked to other administrative datasets. These can then be used as activity indicators: only people who appear on other datasets will be included in the SIDD. The under-coverage of babies can also be addressed by linking to other datasets, as babies can be added in if they appear in other datasets. In particular the birth registration dataset can be used, which covers the registrations of all babies who were born up to the 30th of June 2016. This method is described below.

4. Method Detail

Section 2 listed the broad steps for producing the SIDD. These steps are now explored in detail. The subsection numbers for each stage are the same as the numbering used for each stage in Section 2.

4.1 Standardise the Identifiable Data

Each of the administrative datasets is formatted differently so they are standardised to the variables in Table 2. Names are converted to upper case, and accents are removed from letters at this stage; however other special characters are retained until the linking variables are derived in Step 3.

Table 2 Description of the standardised identifiable variables used.

Variable name	Description
First	First name in upper case letters
Middle	Middle name(s) in upper case letters
Last	Last name in upper case letters
Gender	Gender or sex. Set to 'M' for male, 'F' for female and missing for any other values
Day	Day of birth, stored as a text string of length 2 with leading zeros
Month	Month of birth, stored as a text string of length 2 with leading zeros
Year	Year of birth, stored as a text string of length 4
Postcode	Postcode (unit), stored with no spaces

Not all datasets contain information for all of these variables. For example, the School Pupil Census does not contain names, and Electoral Register (ER) does not contain dates of birth. In addition, datasets may contain further variables that will be used in analysis. A list of these can be found in the Quality Assurance of Administrative Datasets (QAAD).

Note that the information that feeds into these variables may be defined and collected differently in the different administrative datasets. In particular, the gender variable may cover sex from some datasets and gender in other datasets. Some sources may use non-binary values. However, for consistency any values other than male and female are treated as missing for the purposes for linking. This variable is labelled gender in the code, but it is used for the breakdown by sex in the outputs. It is therefore possible that in outputs some people will be recorded as either their sex or gender, if these appear differently on different datasets. This is one of the limitations of combining data from different sources.

In addition, one or two unique identifiers are stored with each record. This allows tracking of individual records across datasets from multiple years. These identifiers

are used when combining the HESA (Higher Education Statistics Agency) datasets for different years, but it is intended that greater use of these will be made to explore internal migration in analysis for future publications.

4.2 Separate Payload and Identifiable Data

At this stage five false records are added to each dataset. These should link to the equivalent records in the other datasets, so if they fail to link this would indicate a problem with linking. These records are removed after the linking stage but before the analysis.

The dataset is given another variable that has a unique value for each record. The dataset is then split into two parts. One part contains the identifiable data used for linking. The other part contains the remaining variables, known as payload. The payload includes gender and age at reference dates (calculated from the date of birth that is included in the identifiable data). These will be used when producing the age–sex breakdowns. Both datasets have the new variable with the unique record identifier. This variable will be used to merge the two parts back together at the National Safe Haven.

The payload dataset is then sent directly to the National Safe Haven, which is managed by the electronic Data Research and Innovation Service (eDRIS). In the case of the Health Activity (HA) dataset a second payload dataset is sent directly from PHS (Public Health Scotland). The dataset with identifiable data is transferred to a separate environment within NRS.

4.3 Generate Linking Variables

The dataset of identifiable data is processed once it is transferred to the separate environment. This is done by another individual. A series of linking variables are derived from the standardised variables listed in Table 2. These are listed in Table 3, and described in more detail in the following subsections.

Table 3 List of derived variables used for linking.

First name	Last name	Postcode	Date of birth / gender
First_alpha	Last_alpha	Postcode	DoB
First_middle	Last_clean	PC_sector	DoB_g
First_clean	Middle_last	PC_district	
First_pt1	Last_pt1	PC_area	
First_pt2	Last_pt2		
First_nickname	Last_mac		
First_nickname2	Last_pt3		
First_drr_clean	Last_drr_clean		
First_5gram	Last_5gram		
First_drr_typos	Last_drrph1		
First_drr_clean2	Last_drr_clean2		
First_drrph1	Last_drr_typos		
First_drrph2	Last_drrph2		
First_drr_typos2	Last_drr_typos2		
First_4gram	Last_drrdm1		
First_drrdm1	End_last_5gram		
End_first_5gram	Last_pt2_5gram		
First_drrdm2	Last_4gram		
First_typo_nickname	End_last_4gram		
End_first_4gram	Last_drrdm2		
First_caver	Last_caver		
First_typo_nickname2	Last_3gram		
First_3gram	End_last_3gram		
End_first_3gram	Last_2gram		
First_2gram	Last_1gram		
First_1gram			

4.3.1 Name Variables

The variables derived from first name are shown in Table 4, and those from last name in Table 5. These variables will be used in linking. Having a range of variables allows links to be made, even if the data has one of a range of errors or corruptions. However, some differences may indicate that the pair of records is a non-match, rather than problems with the data. In order to distinguish between links that are likely to be non-matches, from those that have differences but are still likely to be matches, a distance score is assigned to each link. Larger values of this would indicate that there are greater differences between the identifiable data of the linked records. One of the main components of the distance score is the penalties associated with the linking variables on which the records agree. Further detail on the distance scores is in Section 4.7. The penalties for the various name linking variables are included in tables 4 and **5**. These also show the penalties when there is no agreement on any name variables. In order for larger distance scores to indicate greater differences, the variables that would have the same value even if the names are different are given larger penalties. The penalty values were set in order to best replicate the results of clerical review of links made on test datasets.

Table 4 First name linking variables. The penalty will be added to the overall distance score when the variable is used (see Section 4.7).

Variable	Description	Penalty
First_alpha	First name with characters that are not Latin script removed	1
First_middle	First and middle name combined	2
First_clean	First name with any special characters converted to the most plausible letter (e.g. if first is 'JA#ES' then this would be 'JAMES')	2
First_pt1	The substring of the first name from the start to the first special character (e.g. if first is 'SARAH-JANE' then this would be 'SARAH')	3
First_pt2	The substring of the first name from immediately after the first special character until the next special character (e.g. if first is 'SARAH-JANE' then this would be 'JANE')	3
First_nickname	If the first name is, or has, a common nickname, then this variable is set to a common value for these (e.g. if first is 'ROBERT' or 'BOB' then this would be 'ROBERT'). Also includes variant spellings of the same name (e.g. 'IAN' and 'IAIN').	5
First_nickname2	Similar to first_nickname but with different sets of names (e.g. in the first variable 'SUSAN', 'SUE' and 'SUSIE' are grouped together, while in the second variable 'SUSAN' and 'SUZANNE' are grouped)	6
First_drr_clean	If first name contains a common name as a substring (not necessarily delimited by special characters) then this variable will be set to the longest of these (e.g. if first is 'MARGARETANN' then this would be 'MARGARET')	6
First_5gram	First 5 letters of first name	7
First_drr_typos	Compares, at character level, the first name to names in a name dictionary with the same gender, and calculates similarity (based on substitutions, insertions, deletions, transpositions and jumps) with commonly confused letters (e.g. 'O' and 'D') being considered more similar. Selects the most similar name, with more-common names being given greater weight (e.g. if first is 'SOHN' then this would be 'JOHN').	7
First_drr_clean2	An alternative for first_drr_clean (i.e. the second best matching name)	7

First_drrph1	A phonetic code adapted from the Double Metaphone[2] phonetic coding. Differs from Double Metaphone by not being truncated to four characters, distinguishing between different vowel sounds, and noting vowels at locations other than the start of the string.	8
First_drrph2	The Double Metaphone process produces two codes to capture when some letters can be pronounced differently. This variable uses the second of these codes.	8
First_drr_typos2	As First_drr_typos but uses the second-closest name	9
First_4gram	First 4 letters of first name	9
First_drrdm1	The standard Double Metaphone	9
End_first_gram5	The last 5 letters of the first name	10
First_drrdm2	The second code from the standard Double Metaphone	10
First_typo_nickname	This uses the string identified in first_drr_typos and looks for a nickname root for it in the same way as first_nickname	11
End_first_4gram	Last 4 letters of first name	11
First_caver	Caverphone[3] code of first name	12
First_typo_nickname2	As first_typo_nickname but uses the second nickname	12
First_3gram	First 3 letters of first name	13
End_first_3gram	Last 3 letters of first name	14
First_2gram	First 2 letters of first name	18
First_1gram	First letter of first name	21
	No agreement on first name	27

[2] Double Metaphone was presented in Lawrence (2000).
[3] Caverphone is a phonetic code developed by the University of Otago. See Hood (2004) for further information.

Table 5 Last name linking variables. The penalty will be added to the overall distance score when the variable is used (see Section 4.7).

Variable	Description	Penalty
Last_alpha	Last name with characters that are not Latin script removed	1
Last_clean	Last name with any special characters converted to the most plausible letter as in first_clean	2
Middle_last	Middle and last name combined	2
Last_pt1	As first_pt1 for last name	3
Last_pt2	The substring of the last name from immediately after the first special character until the next special character	3
Last_mac	This will collapse Mac and Mc names to begin with Mac. For example if last name were MACDONALD or MCDONALD then last_mac would be set to MACDONALD to allow these variant spellings to link. Other variant spellings such as THOMPSON and THOMSON are also set to a common value.	4
Last_pt3	The substring of the last name from immediately after the second special character until the next special character	4
Last_drr_clean	As first_drr_clean for last name	5
Last_5gram	First 5 letters of last name	5
Last_drrph1	As first_drrph1 for last name	6
Last_drr_clean2	As first_drr_clean for last name	7
Last_drr_typos	As first_drr_typos for last name	7
Last_drrph2	As first_drrph2 for last name	7
Last_drr_typos2	As first_drr_typos2 for last name	8
Last_drrdm1	The standard Double Metaphone	10
End_last_5gram	The last 5 letters of last name	10
Last_pt2_5gram	The first 5 letters of last_pt2	10
Last_4gram	First 4 letters of last name	11
End_last_4gram	The last 4 letters of last name	11
Last_drrdm2	As first_drrdm2 for last name	11
	No agreement on last name for females born before 2000	11
Last_caver	As first_caver for last name	13
Last_3gram	First 3 letters of last name	14
End_last_3gram	Last 3 letters of last name	15
Last_2gram	First 2 letters of last name	18
Last_1gram	First letter of last name	21
	No agreement on last name	27

Finally, a full name variable (fullname) is generated from a concatenation of first, middle and last name variables (with spaces removed).

4.3.2 Postcode Levels

Postcode area (e.g. 'EH'), postcode district (e.g. 'EH12') and postcode sector (e.g. 'EH12 7') are derived and saved from a postcode variable (e.g. 'EH12 7TF'). As with the linking variables derived from names, these have penalties associated with them corresponding to the level of agreement they indicate. The penalties for the various levels are shown in Table 6.

Table 6 Postcode score assigned to links depending on the level of agreement on address and postcode.

Agreement level	Penalty
Postcode (unit)	3
Postcode sector	8
Postcode district	11
Postcode area	15
One or both postcodes missing	16
No agreement on postcode	19

4.3.3 Date of Birth

The day, month and year variables described in Table 2 are concatenated to generate a date of birth variable called dob. Dob and gender are concatenated to give a single dob_g variable.

To account for similar, but different, dates of birth, Bloom filters are used. These are produced as part of the de-identification step, and so this is described in Section 4.4.2 below (with detail in Annex 2).

4.4 De-identify

4.4.1 Hashing of Identifying Variables

Once these linking variables have had their values defined for each record, the values are transformed in a non-reversible way to a number. This is done with the SHA-256[4] algorithm, a process referred to as hashing. This will always convert the same text to the same number, but even small changes to the text result in completely different numbers.

The text to be hashed has a particular character string appended before being hashed. This appended string is known as the salt and is the same for each record. However, as the value of the salt is held securely, this means that an intruder could

[4] See https://web.archive.org/web/20130526224224/http://csrc.nist.gov/groups/STM/cavp/documents/shs/sha256-384-512.pdf for the specification of SHA-256.

not simply hash the name they were seeking and then search for the resulting hashed value in the dataset. This is because the hashed value of a name will be completely different from the hashed value of the same name with the salt appended.

Hashing does not make the resulting dataset anonymised, as that term is used for cases where the identifiable information is simply removed. However, although the data that is stored is derived from the identifiable data, it is not possible to recover that identifiable information from the data that is stored. The resulting dataset is referred to as pseudo-anonymised. These datasets can therefore be meaningfully linked without using the person's actual name, date of birth, or postcode.

The range of numbers that are output by SHA-256 is larger than those that can be stored precisely in an 8-byte real number. Therefore the number is stored as a hexadecimal string. This hexadecimal string is then truncated to 16 digits to save space, so each hashed variable will take on one of $16^{16} \approx 1.8 \times 10^{19}$ distinct values. With around 10^7 records in the datasets, the probability that there exist a pair of distinct values that end up with the same hashed value (known as a collision) is reassuringly low. Even if there are 5.5 million distinct values, the probability[5] of any collisions is lower than 10^{-6}.

4.4.2 Date of Birth Bloom Filters

For dates of birth, the individual components of day, month and year are hashed using SHA-256. Unlike with names, however, there is no obvious way to correct, or otherwise code, dates of birth, as there will be many similar valid dates of birth. Therefore a method is required to estimate the similarity of different dates of birth from de-identified versions of the date of birth. To do this a coding known as a Bloom filter[6] is used. Further detail on how the Bloom filter is built and used is available in Annex 2.

4.5 Transfer Data to National Safe Haven

Once the dataset of de-identified variables has been prepared it is securely transferred to the National Safe Haven (as described in the DPIA).

4.6 Join Payload and De-identified Datasets

The de-identified dataset is then combined with the payload data for the same dataset. This merge is done using the identification variable that is added to both the de-identified and payload datasets, described in Section 4.2.

[5] See http://www.winlab.rutgers.edu/comnet2/Reading/documents/Birthday_attack.pdf for a description of how the collision probabilities are calculated.
[6] See Schnell, Bachteler and Reiher (2009), for a discussion of using Bloom filters for data linking. Bloom filters were originally presented in Bloom (1970).

4.7 Concatenate Datasets and Link

The datasets from the different sources need to be linked in order to identify which records relate to the same person. This is done by first concatenating the datasets from the different sources into a single dataset. This produces a large data set with every record from each of the source datasets. For example, if there were 8 data sources, each with 5 million records, then the new dataset would have 40 million records. This large dataset is then linked to itself. In this way, links between different sources, and duplicates within individual datasets, can be found at the same time.

This reduces the number of linking steps required. It also avoids problems that arise when linking more than two datasets pairwise. For example, suppose there were datasets A, B and C. Linking these pairwise might involve first linking A to B. Linking to C might involve linking C directly to A. However, this might miss some links, such as a person who is recorded slightly differently on all three datasets, but in a way that meant that the record on B could link to the corresponding records on A and C, but the A and C records would not link with as good a distance score (or perhaps at all). For example, record A might have John Smith, 1/2/1960, B Jon Smith 1/2/1960 and C Jon Smith 2/1/1960 (although recall that by the linking stage the data will be de-identified). Datasets B and C could also be linked. As the number of datasets increases the number of pairwise comparisons increases at a greater rate, so with four datasets there would be six pairwise comparisons, with five datasets there would be 10 pairwise comparisons, following the pattern of 3, 6, 10, 15, 21 and so on (triangular numbers). Concatenating all the datasets allows all the links that would be found by doing all the possible pairwise steps to be found in just one step. This provides information on the similarity of the records in datasets A, B and C, which can be used to decide whether these represent the same person or different persons.

The linking is done by performing a series of iterations. Each iteration uses a different matchkey. At the start of each iteration the matchkey value for each record is produced. This is generally done by concatenating the values from a number of linking variables. For example a matchkey might have first name, last name and postcode. In such cases the matchkey would take values such as 'JOHN_SMITH_EH127TF'. The dataset is then sorted by this matchkey and linked to itself. In this way, all the records that have exactly the same value of the matchkey are found. The matchkeys used in the linking are shown in Table 7. The matchkey is used as a blocking[7] variable, where every pair of records that have the

[7] When blocking, the records for linking are separated into blocks with the same value of some blocking variable(s). Links are only sought within (rather than between) blocks. There will then be no links where the linked records have different values for the blocking variable(s). See Steorts et al. (2014) for a discussion of blocking.

same value of the matchkey are compared. For example, postcode, first_alpha and last_alpha are used for Matchkey 4. An example of this might be EH127TF_SARAHJANE_OCONNOR (note the removal of non-letter characters from the names). All the records with this value would then be compared with this record, as well as with each other.

Table 7 List of matchkeys used in the linking. If date of birth is indicated as Bloom then the Bloom filters are used in addition to using date of birth exactly. 'General' on first and/or last name means that all the first and/or last name linking variables are considered. The penalty of the linking variable with the lowest penalty where the records still agree is what is used.

Number	Matchkey (blocking variables)	DoB	Note
1	PC_district	Exact	General on first and last name
2	Postcode first_alpha	Bloom	General on last name
3	Postcode last_alpha	Bloom	General on first name
4	Postcode first_alpha last_alpha	Missing	
5	First_alpha last_alpha	Exact	Missing postcode
6	PC_area first_alpha	Exact	General on last name
7	PC_area last_alpha	Exact	General on first name
8	First_alpha last_alpha	Exact	
9	PC_sector first_alpha last_alpha	Bloom	

For some matchkeys, date of birth must agree exactly for a pair of records to be considered. This effectively makes date of birth another blocking variable. For other matchkeys, (indicated by 'Bloom' in Table 7) the matchkey is used twice. The first time the date of birth is used as part of the matchkey, so that date of birth must agree exactly. The second time, date of birth is not included in the matchkey, but the Bloom filters are used to compare the similarity of date of birth. This is done by counting the number of bits that differ between the two Bloom filters (see Annex 2 for further details).

For a link to be made between records for Matchkey 4, the date of birth must be missing on one or both of the records. A penalty of 12 is added to the overall distance score for date of birth being missing. Similarly, on Matchkey 5, a link is only made if postcode is missing on one or both of the records.

For each link that is made a distance score is calculated (where lower scores indicate stronger links). This is found by summing the penalties of the linking variables that make up the matchkey (see tables 4 to 6). If the Bloom filters are used for the date of birth then an additional penalty is added according to how many bits differ. The penalty is given using the recurrence relation shown in Table 8. As with other penalties, this was chosen to best fit judgements from clerical review of links from test datasets. Even if no bits differ on the Bloom filters a penalty of 1 is incurred. This is because there may be differences in the date of birth, but by chance they ended up with the same Bloom filter. In cases where the dates of birth

are exactly the same then the records would link using one of the matchkeys that included date of birth (and no penalty would be added for date of birth).

Table 8 Penalty ($\boldsymbol{P}$) for differences between Bloom filters of $\boldsymbol{n}$ bits. The penalties for differences of 2 or more bits are calculated using the recurrence relation (where $\lceil \boldsymbol{x} \rceil$ is the ceiling function of $\boldsymbol{x}$).

n	$P(n)$
0	1
1	2
2–32	$P(n-1) + \left\lceil \frac{n-1}{2.5} \right\rceil$

When the matchkey indicates that first name is 'general' then agreement is not required on first name. Instead records that agree on the other blocking variables are compared and the level of agreement on first name is measured. This will compare the hashed variable values for the linking variables listed in Table 4. A penalty is added to the distance score equal to the penalty associated with the linking variable that has agreement between the two records, and has the lowest penalty. An equivalent process is used for 'general' last names.

If the value of the matchkey that is used to link the records is not especially rare then an additional penalty is added to the distance score (as in these cases it is more plausible that records may relate to different persons). This is done by finding the number of occurrences of each value of the linking variables used in a matchkey. So if the matchkey used first_alpha, last_alpha and postcode then the counts for each value of these variables are identified[8]. Each count is then divided by the approximate population, to give a probability that any random person will have that value for the linking variable. The probabilities for all the linking variables used are multiplied together, and then multiplied by the approximate population to give an expected number of people in Scotland with the particular combination of linking variable values (e). If this number is less than 2^{-10} then no modification is made to the distance score. If the number is greater than or equal to 4 then the link is discarded. Otherwise, $10 + \lceil \log_2 e \rceil$, where e is the expected number (and $\lceil x \rceil$ is the ceiling function of x), is added to the distance score.

Once the linking has been completed for a particular iteration, another iteration is processed using the next matchkey (made up of different linking variables). All the links made in this process are stored in a dataset known as the link repository.

[8] The counts are the number of times each value appears on the NHSCR.

4.8 Trim Links

At this stage the link repository is trimmed down to a set of links to be used in further analysis. This removes any link with a distance score greater than 25. The threshold of 25 was chosen because clerical review of links from a previous test indicated that links beyond that score started becoming unreliable. However, the effect of different choices of this threshold will be explored when reviewing the method in advance of the next publication.

Note that Matchkey 4 will be particularly useful when linking records from the Electoral Register datasets, as date of birth is not included on them. Similarly, Matchkey 1 will be particularly useful when linking records from the School Pupil Census dataset as names are not included on that dataset. Matchkey 1 uses the 'general' linking for first and last name, and one of the records having missing values is one of the considerations. Note that these links will typically get large distance scores. In order to avoid links for records in these datasets being discarded for being too weak, manual adjustments are made to the overall distance scores of links involving such records. Links with records from the School Pupil Census with an overall distance score of 35 (indicating exact agreement on postcode and date of birth, and first and last name missing) are rescored as 25, ensuring that they are not discarded. Links with records from one of the electoral register datasets that were made using Matchkey 4 (missing date of birth) have their distance score reduced by 3, also ensuring that such links are not discarded.

The reference date used was the 30th of June, in order to be comparable with the published MYE. As the reference date lies between academic years, two datasets are used for HESA: 2015/16 and 2016/17. In addition, each record on the HESA dataset is duplicated, one showing the person at their term-time postcode, the other at their home postcode. This means that each higher-education student could appear in the data four times. In order to ensure that these records are all grouped together, the HESA datasets are analysed and the records grouped by the original IDs (hashed HESA ID and Scottish Candidate Number). The links involving any of the HESA records in the group are then identified. These links are then changed so that they all link to the same HESA record in this group. This might result in some links being duplicated (in cases where the different HESA records for a person get linked to the same record from another dataset), so these are then de-duplicated.

At the later analysis (see Section 4.9) when records are resolved to people, it will be likely that the one HESA record that links to the other records will be given the same Unique Personal IDentifier (UPID, see Section 4.9.2 for further information on UPIDs) as some records from other datasets. The remaining HESA records will all get their own UPID, as they will not have any links. However, at that stage the UPIDs of the isolated HESA records are changed, so that they get the same UPID as the HESA record that is linked. This approach ensures that all the HESA records

with the same ID get the same UPID, and that this is the same as any relevant records from other datasets.

Table 9 Number of links kept after the trim links stage, broken down by the distance score.

Distance Score	Number of Links
5	13,913,357
6	1,193,418
7	51,804
8	21,007
9	226,621
10	346,568
11	287,047
12	33,293
13	224,010
14	133,101
15	6,510,495
16	1,063,272
17	1,589,575
18	1,274,858
19	535,594
20	355,435
21	353,888
22	180,839
23	77,631
24	49,590
25	1,782,276
Total	30,203,679

The number of links by the distance score that are kept after the trim links process is shown in Table 9.

4.9 Resolve Linked Records to Individuals

4.9.1 Assign Records to Components

There will be more records across the administrative datasets than there are people these represent, as many people will appear in more than one dataset. Therefore to find out how many distinct people appear on the datasets, the links found are used to resolve the records down to distinct persons.

In some cases a group of records will all link to each other, and not link to any records not in that group. In these cases the records can be straightforwardly resolved into a single person. Other groups of records may not have all records linking to each other, but they still cannot be separated into distinct groups without separating linked records. In these cases an algorithm is used to determine which records should be considered the same person. This tends to assign records to different persons if no direct link could be made between them.

To analyse the data it is considered as a graph[9], where the records are the vertices and the links are the edges. The algorithm starts by assigning all the records a unique number ID. Then records that link to each other, either directly, or via intermediate records, are assigned to the same component[10]. (A component of a graph is a subgraph where there is a path[11] between any pair of vertices in the component, but not paths to any other vertices. For example, in Figure 4 there are three components, with two, three and four records.) Both records will take the lower of the number IDs of the two records. This is done by performing a depth-first search[12] of the records. By the end of this search, all the records in the same component will have the same ID, and all the records in different components will have different IDs.

For each component the number of records is obtained, along with the number of records from each dataset, and the number of links within the component.

4.9.2 Assign UPIDs for Complete Components

Finally, each record is assigned a Unique Person IDentifier (UPID). Records that are believed to represent the same individual are assigned the same UPID. Conversely, records estimated to represent different individuals will have different UPIDs.

To assign UPIDs, components are considered separately. Records in different components are not linked directly or indirectly. Therefore, these should be considered different persons, and be assigned different UPIDs. Thus, when starting on a new component, none of the UPIDs that have been used for different components should be used. For example, in Figure 4 there are three components, and none of the records in each component have a UPID (displayed here by the colours) that appears in any of the other components. These particular components are complete[13], in that every record links to every other record in the component.

[9] See https://mathworld.wolfram.com/Graph.html for a definition of a graph.
[10] See https://mathworld.wolfram.com/ComponentGraph.html for a definition of a component.
[11] See https://mathworld.wolfram.com/GraphPath.html for a definition of a path.
[12] See https://mathworld.wolfram.com/Depth-FirstTraversal.html for a description of a depth-first search.
[13] See https://mathworld.wolfram.com/CompleteGraph.html for a definition of a complete graph.

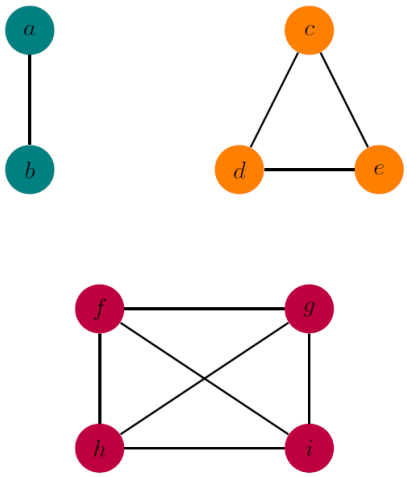

Figure 4 A set of records (represented by vertices) and their links (represented by edges) in three complete components. Letters indicate the record ID. Colours represent the record UPID. Examples of what these records might represent are given in Table 10.

If a link is not found between two particular records then it is assumed that these represent distinct persons. This approach is taken partly because the search for links is fairly thorough, allowing for differences in name, date of birth and location. Therefore, if the process did not find a link between the records then it is supposed that the records actually represent different persons. The other reason for this is that even if it results in records for the same person getting different UPIDs, then one of them can be discarded when business rules are applied (see Section 4.10 for details on this). For example, there is a business rule to only include UPIDs where there is a record from NHSCR (apart from babies). Therefore, if a record from a different dataset failed to link then it will be discarded, and not be counted as an extra person in the SIDD.

Table 10 Example (mock) data for the records used in Figure 4. (Note that by the time the records are linked these data would be hashed.)

Label	Name	Date of Birth	Postcode	Dataset	UPID
a	Kenneth MacKenzie	1/1/1960	EH1 1AA	NHSCR	1 (green)
b	Kenny MacKenzie	1/1/1960	EH1 1AA	Health Activity	1 (green)
c	Maya Patel	2/3/1950	AB1 1AA	NHSCR	2 (orange)
d	Maya Patel	2/3/1950	AB1 1AA	Health Activity	2 (orange)
e	Maya Patel		AB1 1AA	Electoral Register	2 (orange)
f	Olivia Wilson	4/5/1970	G1 1AA	NHSCR	3 (red)
g	Olivia Wilson	4/5/1970	G1 1AA	Health Activity	3 (red)
h	Olivia Wilson		G1 1AA	Electoral Register	3 (red)
I	Olivia Wilson	4/5/1970	G1 1AA	Marriages	3 (red)

Assigning UPIDs to records within a component can then be thought of as a vertex colouring[14] problem. As before, the graph would be made up so that records would

[14] See https://mathworld.wolfram.com/VertexColoring.html for a description of vertex colouring problems.

be the vertices and the links would be the edges. Each colour assigned to a vertex would represent a different UPID, a different person. However, normally in vertex colour problems linked vertices get assigned different colours (while unlinked records might get the same colours). In this case it is the unlinked records that need different colours (UPIDs). Therefore, the UPID allocation can be thought of as a vertex colouring on the complement of the component (where unlinked vertices are linked and vice versa[15]).

Thus, if there are some different components, and each record in the component linked to each other record in the component (that is, each component is complete, see Figure 4), then all the records in the component should be assigned the same UPID (represented in Figure 4 as the vertices in the same component getting the same colour).

4.9.3 Incomplete Components

The process is more complicated when there are records in the component that are not linked, as in Figure 5. In this case record *a* and record *c* are not linked, so they need to be assigned different UPIDs. This is represented in Figure 5 by the records getting different colours. Record *b* links to record *a* and record *c*, but as they have different UPIDs *b* cannot get the same UPID as both of them. However, it is likely that record *b* represents the same person as one of *a* or *c* does, so it is desirable that *b* be assigned the same UPID as either *a* or *c*. In order to decide which, the distance scores of the two links are considered. The link between *b* and *a* has a lower distance score than the link between *b* and *c* (indicating *b* is more similar to *a* than it is to *c*). Therefore *b* is assigned the same UPID as *a*.

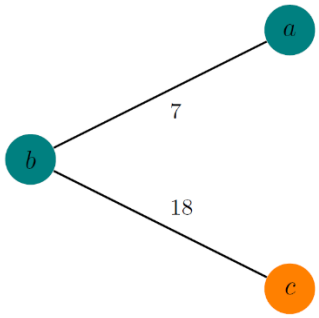

Figure 5 A set of records (represented by vertices) and their links (represented by edges) in one incomplete component. Letters indicate the record ID. Colours represent the record UPID. Numbers indicate the distance score of the link between the records.

It can be seen from this how the process is similar to a vertex colouring of the complement. In the complement the only link would be between *a* and *c*, so these would need to be assigned different colours. As *b* is linked to neither *a* nor *c* then *b*

[15] See https://mathworld.wolfram.com/GraphComplement.html for a definition of the complement of a graph.

can take the same colour as one of them, and doing so will minimise the number of colours needed.

Table 11 shows how many components of various sizes are complete (like in Figure 4) or not complete (like in Figure 5) were generated from the 2016 data. Most of the small components are complete, but more of the larger components are not complete.

Table 11 Number of components identified, by the number of records in the component and whether it is complete (that is, whether all records link to all other records in the component).

Number of records in component	Component complete?	
	No	Yes
2	0	931,234
3	64,352	1,651,560
4	233,744	2,328,157
5	201,014	344,188
6	116,249	43,476
7+	105,756	5,162
Total	721,115	5,303,777

4.9.4 Initial UPID allocation

The first step in allocating UPIDs to achieve the results described in subsections 4.9.2 and 4.9.3, is to allocate UPIDs to records that are not linked to records that already have UPIDs allocated. The algorithm to do this is given below:

- Loop through vertices (i) in component
    - Loop through vertices (j) in component (where $i \neq j$)
        - If i is linked to j, and j has a label then:
            - Flag i to not be labelled
    - If i has not been flagged then assign it a new UPID

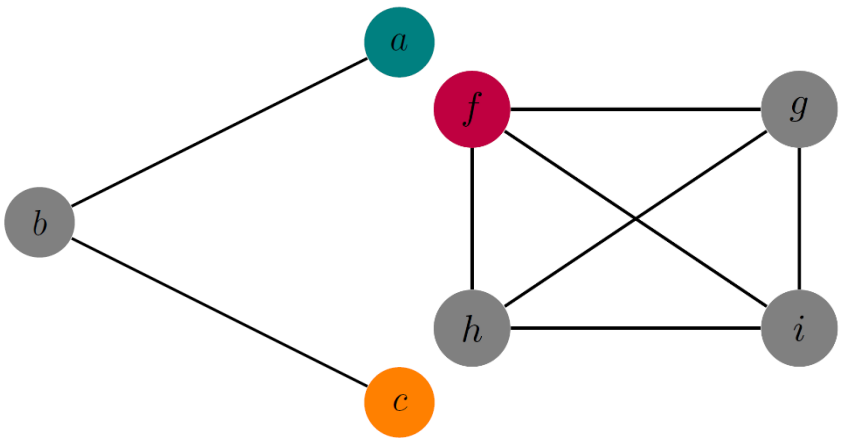

Figure 6 A set of records depicted as in Figure 5 (grey represents records that have not been allocated a UPID in the initial allocation).

This will label between 1 and $\left\lceil \frac{n}{2} \right\rceil$ vertices, where n is the number of vertices in the component. Figure 6 shows two examples of how the initial UPID allocation might go. In the first component record *a* is allocated a UPID first. As record *b* is linked to *a* then it is not allocated a UPID. Record *c* only links to an unlabelled record (*b*) so it gets a new UPID. In the second component record *f* is allocated a UPID first. All the remaining records link to *f* and so are not allocated a UPID at this stage.

4.9.5 UPID Allocation to Remainder

Following the initial UPID allocation, another algorithm is performed to allocate UPIDs to the remaining (unlabelled) records. This tries to assign UPIDs in such a way that records get the same UPID as that of the records to which they link with the lowest distance score. This algorithm is as follows:

- Loop through distance scores (*s*), lowest first (5 to 25)
    - Loop through unlabelled vertices (i)
        - Loop through vertices (j) in component (where $i \neq j$)
            - If i is linked to j at score *s* or lower and j has a label then:
                - Allocate i the same UPID as j (unless one of the exceptions described in Annex 3 apply)

If all distance scores have been looped through and some records still have not been allocated a UPID then they will each get allocated a different UPID that has not been used for any other records.

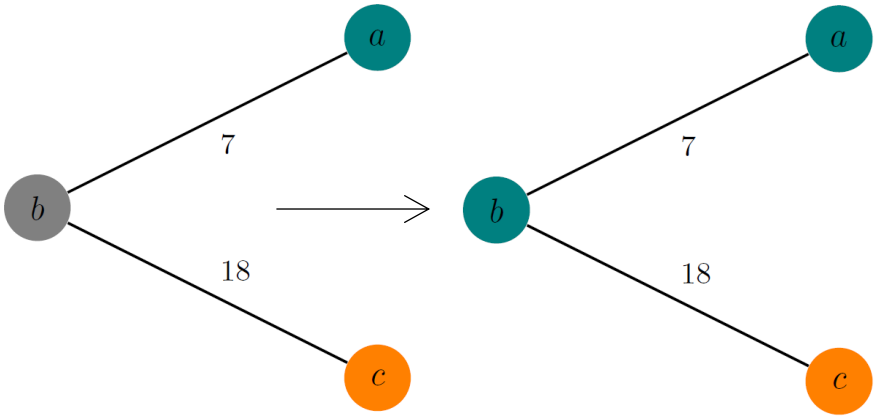

Figure 7 Demonstration of the UPID allocation to a component following the initial allocation. The records and links are depicted as in Figure 6. The graph on the left shows the situation following initial allocation. The graph on the right shows the situation after the remainder have been allocated.

Figure 7 shows how this might happen. Following initial UPID allocation records *a* and *c* have been allocated (different) UPIDs. Looping through the distance scores the algorithm starts with no links, and then the link between record *a* and *b* appears

at the distance score of 7. At this point record *b* will be allocated the same UPID as record *a* has (as *b* links to record *a*). By this point all the records have had UPIDs allocated, and so the algorithm can halt. There are some cases, however, where the above algorithm gives an undesirable result. Therefore, modifications are needed in the form of exceptions. These exceptions are described in Annex 3.

Once every record from the source datasets has been allocated a UPID, a single dataset of all these records with their UPIDs is set up. This dataset is referred to as the Administrative-Data Record Set (ADRS).

4.10 Apply Business Rules

Having resolved the records, an estimate of how many distinct persons appear on the administrative data is available, that is, the number of distinct UPIDs. However the number of persons appearing on the administrative datasets will not necessarily be an accurate count of the population. There are several reasons for this:

- Some of the persons who appear on one or more datasets are not present in the population at the reference date.
- There will be persons who are in the population but do not appear in the administrative datasets.

For some persons there is information on the administrative datasets to indicate that they are no longer in the population. Data from vital events or NHSCR can indicate that a person has died. These records are kept in the datasets so that if they link to records in other administrative data records then they can be excluded. Similarly, the NHSCR and the Electoral Register can indicate that a person is no longer living in Scotland, and the records for these persons can also be removed.

A new dataset is built from the ADRS. This will have one record per UPID, and will be trimmed down using business rules to only those UPIDs that it is estimated represent persons who are present in the Scotland population on the reference date. This new dataset is Scotland's Integrated Demographic Dataset (SIDD). The counts of persons on this dataset are then used as the population estimate: the Administrative Data Based Population Estimate (ABPE).

4.10.1 Addressing Under-coverage of Babies in NHSCR

Section 3 showed that at any time there will be some babies in the population who are yet to appear on the NHSCR (mainly because their births are yet to be registered). Some babies may have appeared but without a posting, so including the babies who are on NHSCR but do not have a posting improves the estimate somewhat, but there remains under-coverage. To address the issue of under-coverage of babies, the birth registrations data is used, as that includes the registrations of births that take place up to the reference date. After linking the birth registrations data to the NHSCR, any person who appears on the birth registrations,

and is zero years of age, is included, even if they do not appear on the NHSCR or appear on the NHSCR with a missing posting.

4.10.2 Addressing Over-coverage in NHSCR

Section 3 also showed that NHSCR has over-coverage for age groups other than babies. The first step is to remove records where there is an indication that they are no longer in the Scotland population. Some are removed using the NHSCR postings discussed in Section 3. In addition, records are also removed if the Electoral register indicates that they are an overseas voter, or if they have a death registration with the death date on or before the reference day.

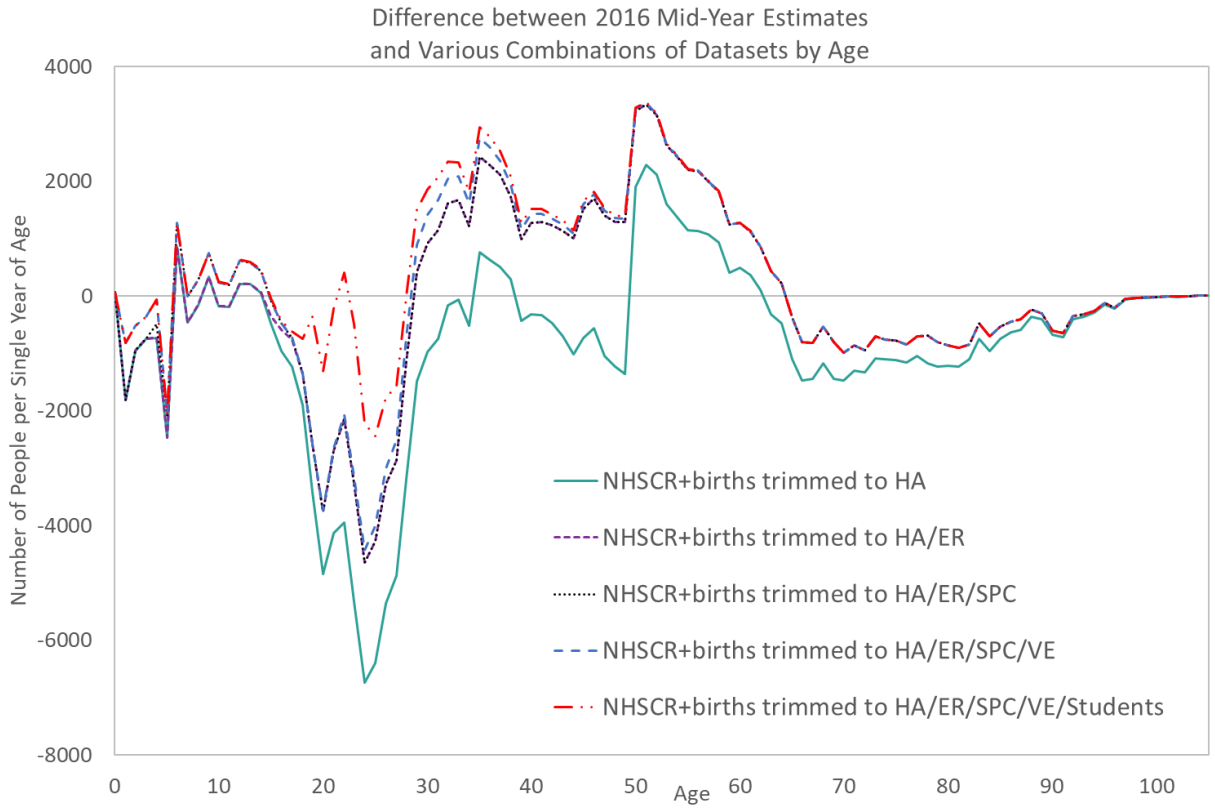

Figure 8 Build up to the SIDD by adding activity datasets. HA relates to Health Activity, ER Electoral Register, SPC School Pupil Census, VE Vital Events and students HESA and FES.

To address the remaining over-coverage, the records for anyone who does not also appear on another administrative dataset are removed. Figure 8 shows how cutting the NHSCR down to combinations of the various activity datasets affects the population estimates. Firstly, the Health Activity dataset is considered. This is a dataset of any NHS patient who has interacted with the NHS in the three years prior to the reference date. Cutting the NHSCR down to the Health Activity dataset removes the over-coverage for most age groups. A notable exception is from age 50 up to age 62. The jump at age 50 might be because this is the age that the bowel national screening programme starts, and so many patients will have increased

interactions with the NHS at this age. Therefore, more patients will appear on the Health Activity dataset around this age.

For most ages trimming the data down to people who appear on the Health Activity dataset results in under-coverage, so further activity datasets are required. The next step is to include the Electoral Register (and trim NHSCR down to people who appear on Health Activity *or* Electoral Register datasets). This increases the estimates for all ages 15 and above. This results in over-coverage for ages 29–64. Above age 64 there remains under-coverage, and adding the other available datasets does not noticeably improve this. There is also under-coverage for pre-school children (apart from babies, which have been treated separately) and 15–28 year olds.

Adding the school pupil census (SPC) dataset helps with the under-coverage of the 15 year olds (and to a lesser extent 16 and 17 year olds), although leads to some over-coverage for younger school-age people.

To address under-coverage for pre-school children vital events data can be added. This includes the birth registrations mentioned above (going back five years), but also the registered parents and marriages and civil partnerships. This helps with the under-coverage of pre-school children, although increases the over-coverage among middle-aged people (who may become parents or enter marriages and civil partnerships).

The remaining substantial under-coverage is for people aged around 18–27. To help address this the HESA and FES datasets are added. This substantially reduces the under-coverage for that group, although some under-coverage remains, especially for people in their mid-20s.

Therefore, the final business rules for which UPIDs are excluded from or included in the SIDD are given below.

If any of the following conditions are met then the UPID is excluded from the SIDD (even if the inclusion conditions are met):

- UPID appears on the death registrations with a death date on or before 30/6/2016
- UPID appears on the NHSCR with one of the categories indicated above to be excluded (embark or outwith Scotland)
- UPID appears on one of the electoral register datasets where the franchise is ‘F’, indicating that the voter lives overseas.

If none of the above conditions are met then the UPID will be included on the SIDD if at least one of the following conditions are met:

- UPID appears on NHSCR with one of the categories indicated above to be included (that is, a Scottish posting) and at least one other dataset (not counting Registers of Scotland)
- UPID appears on the birth registrations as a child and is aged below one.

Registers of Scotland data was not used as an activity indicator as when people buy a property they may not move to that location, for example if they are buying to let.

4.11 Assign Analysis Variables

If age is available from the NHSCR then that age is used. If not, then age is taken from one of the other datasets where it is not missing. The prioritisation for this is given in Table 12. All records that passed the condition for inclusion in the SIDD had an age available. Gender is treated the same way. Records are excluded from the SIDD if gender is still missing, although this only affected an extremely small number of records. The outputs are presented by sex, where sex is taken from the gender variable. As discussed above, this variable may contain information on sex or gender, depending on the data source from which it was taken. NHSCR is the data source for most records.

Postcode is taken from the NHSCR if it is available there. If not, then it is taken from one of the other datasets where it is not missing. The datasets with the highest priority are used first, and the others are used in order of prioritisation set out in Table 12, if information is not available from any of the preceding datasets. However, the postcode gets overwritten for students if the UPID appears both on HESA 2015/16 and 2016/17 and the term-time postcode is the same on each of these, in which case the term-time postcode is used.

Table 12 Prioritisation of assigning postcode, age and gender.

Dataset	Priority
NHSCR	1
Health Activity	2
School Pupil Census	3
Electoral Register	4
Registers of Scotland title	5
Births	6
Civil partnerships	7
Deaths deceased	8
Deaths informant	9
HESA 2016/17	10
Further Education Statistics	11
Marriages	12
HESA 2015/16	13

There was an intention to also use information from the Registers of Scotland (RoS) dataset to resolve where there are conflicting postcodes between two different datasets. If a record had different postcodes from different sources, and one of the

postcodes agreed with the postcode from a linked RoS record, then the moving date from RoS would be compared with the reference date. However, no such cases were found in the data. Whether this is a problem with the analysis, or whether such cases are simply so rare that none were found, will be investigated in the future.

A lookup table from hashed postcode to council area, urban–rural classification (2013/14) and SIMD decile (2016) is used to assign each SIDD record to each of these geographies. For some records no valid postcode is available. These cases are shown in the tables with missing geography, in order to show how many such cases there are.

5. Future Plans

The counts presented here have a mixture of over-coverage and under-coverage. Some over-coverage happens even when just using the Health Activity dataset as the only activity dataset. Reducing the size of the Health Activity dataset could address this. This could be done by including patients who have activity within a period of time smaller than three years, perhaps one year, or six months. It may be that different thresholds could be used for different demographic groups, depending on how frequently those groups typically interact with the NHS.

Such modifications would address over-coverage, but would likely result in greater under-coverage. However, there is no particular reason that the ABPE must be the counts of persons on the SIDD. Estimation methodologies, such as dual-system estimation[16], could be used to produce the ABPE from the SIDD, addressing under-coverage in the SIDD. Dual-system estimation requires two distinct lists of people to be linked together. This might involve taking a SIDD, similar to what is presented here, and linking that to data from a coverage survey. Another option would be to make up the two lists using different administrative datasets. Each would need to cover all age groups, as estimation would need to be stratified by age. One option would be to use Health Activity as the first list, and the combination of vital events, School Pupil Census, HESA and FES, and Electoral Register for the second. Each list would then cover all age groups (vital events would cover pre-school children, School Pupil Census would cover school-age pupils, Electoral Register would cover adults, and HESA and FES would top up the 20s age group). These lists could also be trimmed down to active NHSCR records, in order to avoid including people who have left the country. When this is done using the 2016 data (with the full Health Activity dataset), comparable results are obtained as using the SIDD on its own. However, using this method affords greater flexibility, making it easier to deal with both over and under-coverage. It is intended that is area will be investigated in the future.

[16] Dual-system estimation is used in Scotland's Census to address under-coverage. See the Estimation and Adjustment Methodology paper for more information.

6. References

Bloom, B. (1970) ‘Space/time trade-offs in hash coding with allowable errors’, *Communications of the ACM*, vol. 13, no. 7, pp. 422–426

Hood, D. (2004), *Caversham Project Occasional Technical Paper*, [Online] available at: http://caversham.otago.ac.nz/files/working/ctp150804.pdf (accessed 31 August 2020)

Lawrence, P. (2000), ‘The double metaphone search algorithm’, *C/C++ Users Journal*, vol. 18, no. 6, pp. 38–45

Schnell, R., Bachteler, T. and Reiher, J. (2009) ‘Privacy-preserving record linkage using Bloom filters’, *BMC Medical Informatics and Decision Making*, vol. 9, no. 41

Steorts, R., Ventura, S., Sadinle, M. and Fienberg, S, 2014 ‘A Comparison of Blocking Methods for Record Linkage’ in: Domingo-Ferrer J. (eds) *Privacy in Statistical Databases: Lecture Notes in Computer Science*, vol. 8744

Annex 1: Glossary

Term	Definition
ADRS	Administrative Data Record Set. Once the source datasets have been linked, the ADRS contains a record for each individual who appears on these source datasets.
Collision	With reference to hashing, a collision is when two different strings get hashed to the same number. With reference to Bloom filters, a collision is when the same bit is set to 1 more than once (rather than multiple bits getting set to 1).
EROs	Electoral Register Officers (www.saa.gov.uk/electoral-registration).
HA	Health Activity dataset.
Hash	Converting text to a number in a non-reversible way. This is done in such a way that small changes to the text are highly likely to result in a different number. The hashing algorithm used here is SHA-256.
HESA	Higher Education Statistics Agency (www.hesa.ac.uk).
Hexadecimal	Base 16 numbers. Digits 1–9 are used for 1–9 and then letters A–F are used for 10–15.
Link	Two records that have been connected.
Match	Two records that represent the same individual.
Modular Arithmetic	In modular arithmetic the remainder is obtained when dividing a number by a modulus. For example, using a modulus of 32 means that 34 mod 32 = 2.
MYE	2016 Mid-Year Population Estimates for Scotland (www.nrscotland.gov.uk/statistics-and-data/statistics/statistics-by-theme/population/population-estimates/mid-year-population-estimates/archive/mid-2016).
NHSCR	National Health Service Central Register. A dataset of people registered with an NHS GP in Scotland, or who were born or died in Scotland.
Non-match	Two records that represent different individuals.
PHS	Public Health Scotland (https://publichealthscotland.scot).
ROS	Registers of Scotland (www.ros.gov.uk).
SAS	The statistical software used to process the datasets and run the linking (www.sas.com).
Salt	A character string appended to variable values before they are hashed. This is the same for each record and is held securely. This avoids any intruder being able to find the record for an individual they knew by hashing the name and searching for that hashed value in the dataset.
SFC	Scottish Funding Council (www.sfc.ac.uk).

SGLD	Scottish Government Learning Directorate (www.scotland.gov.uk/Topics/Statistics/Browse/School-Education/Summarystatsforschools).
SIDD	Scotland's Integrated Demographic Dataset. This is similar to the ADRS, but only contains records that have passed the business rules (that is, records for which it is believed that the person they represent is present in the population at the reference date).

Annex 2: Date of Birth Bloom Filters

First the date of birth is separated into components: day of month, month, decade and individual year. For example, 12/02/1985 would be '12', '02', '8', '5'. Gender is also used, so that comparisons on gender can be made while comparing the Bloom filters. Each component value then has a salt appended to the end of the string. The new string (the component plus the salt) is then hashed, using the SHA-256 algorithm discussed above. The first two hexadecimal characters of the SHA-256 output are identified and converted to a number, which will be an integer between 0 and 255. Then, using modular arithmetic, this number modulo 32 is taken, and 1 added, to give a number between 1 and 32. This number is then used to set the corresponding bit in a 32-digit[17] binary number to 1. For example, in Figure 9 the red arrows show that the 18th bit is set to 1 for two similar dates of birth. (Note that multiple components might point to the same bit. If this happens the bit will still be set to 1, it does not get toggled back to 0.) In this way a binary number, known as the Bloom filter, can be built up (see Figure 9).

Although this binary number is derived from the date of birth, it would be practically impossible to recover that date of birth from the Bloom filter. However if dates of birth are the same on different records then they would have the same value in the Bloom filter. More importantly though, if two dates of birth had similar digits, for example 12/02/1985 and 12/02/1995, then the Bloom filters would also be similar in that it would expected that most of the bits in the Bloom filters have the same value. The similarity of dates of birth can therefore be measured by counting the number of bits that are the same in the Bloom filters.

[17] 32-digit numbers are used as these are the largest numbers that bitwise operations can be run on in SAS.

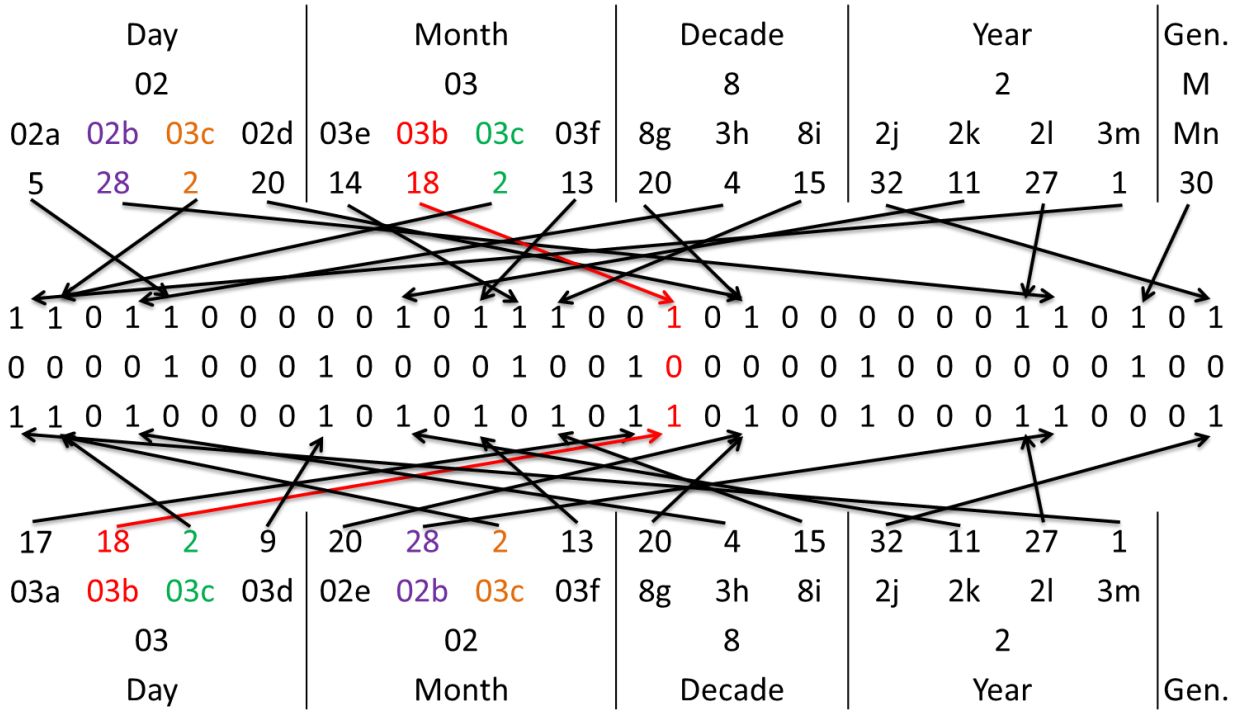

Figure 9 Demonstration of production of Bloom filters for two records, and the comparison between them. The encoding of one record (a male, with date of birth 02/03/1982) is made from the top, while the other (date of birth 03/02/1982 with missing gender) is shown from the bottom. The adjacent rows show the text that gets hashed after the component has had digits collapsed and had the salt added. The next rows show the number of the bit that is to be set to 1, and the arrows show where these are in the binary number. The middle row shows the bitwise comparison of the two binary numbers (0 indicates they are the same, while 1 indicates they are different).

Each component contributes to the Bloom filter multiple times. This increases the number of possible Bloom filters. For example, only using one bit would mean that there could only be 32 possible Bloom filter values. In general the number of Bloom filter values with k bits set to 1 is given by:

$$\binom{32}{k} = \frac{32!}{k!\,(32-k)!}$$

This is maximised when $k = 16$. However, it is possible that the number of bits that are set to 1 in the Bloom filter will be less than the number of times a bit is set to 1. This is because multiple contributions may refer to the same bit (referred to as a collision). This means that the maximum amount of information could be stored if more than 16 contributions are made to the Bloom filter, although not so many that it is likely that most of the bits are then set to 1. However, the more contributions that are made the more collisions there will be. In addition to maximising the amount of information stored, it is also desirable to avoid too many collisions. Components whose contributions collide would result in smaller differences when Bloom filters are compared than components whose contributions do not collide, so this affects the balance of the information from the various components. To roughly balance these

effects it was decided that a total of 16 contributions should be made to each Bloom filter.

Table 13 Number of times each component is salted and makes a contribution to the Bloom filter.

Component	Number of contributions to Bloom filter
Day	4
Month	4
Unit year	4
Decade	3
Gender	1
Total	16

The number of contributions from each component to the Bloom filter are shown in Table 13. Gender only has one contribution in order to give much greater weight to date of birth. This is desirable as the gender variable has fewer possible values (three, including missing) and the reliability of this information can be lower. Decade has slightly fewer contributions than the other components as people's dates of birth are spread less evenly across decades than across individual years within decades (for example, there are fewer people who were born in the 30s than in the 90s).

With each component making multiple contributions, each contribution can be treated slightly differently in order to tune the comparisons. This would make differences that are caused by common corruptions more similar than differences that are more likely to represent an underlying difference. For example, sometimes a date of birth is recorded in American format by mistake (that is, in MM/DD/YYYY instead of DD/MM/YYYY). It is therefore desirable that the Bloom filter for 12/02/1985 be quite similar to (although not exactly the same as) that for 02/12/1985. To achieve this, two of the four salts that are used for the day component contributions are the same as two of the salts used for the month component contributions. This will mean that two dates within a specific year that could be different formats of the same date will on average be half as different as two dates that are completely different (within the same year).

Having multiple contributions also means that some pairs of digits can be considered less different than others. For example, a '1' and a '7' can be confused if a hand-written form is scanned or typed in, but a '1' and a '2' are less likely to be confused. Thus, before adding the salts, for some steps a '7' value is changed to a '1', while keeping these different when using the other salts. This would mean that dates of birth that have a '1' and a '7' for a component (for example, unit of year) will still result in different Bloom filters, but not (typically) as different as dates of birth that have values of '1' and '2'. As each component makes multiple contributions to the Bloom filter, the digits can be collapsed in different ways for each contribution. This

allows differences between collapsed differences to still be detected through other contributions.

In this way differences that relate to common scanning errors can be given smaller penalties than other differences. Doing this separately for different contributions means that the collapsing of digits need not be transitive. For example, ‘5’ is collapsed with ‘4’, and also ‘5’ with ‘6’. But by doing these separately means that ‘4’ need not be collapsed with ‘6’, which is what is desired, because ‘4’ and ‘6’ are less easily confused.

Annex 3: Exceptions Used in the UPID Allocation Algorithm

There are some cases where the algorithm that assigns the remainder of the UPIDs gives an undesirable result. Therefore the modifications described below are applied to it.

Exception 1

Figure 10 gives an example of when exception 1 would need to be applied. Here record *b* has been assigned a UPID in the initial allocation. In the algorithm for allocating UPIDs to the remainder, record *a* will first be allocated a UPID, the same as that for record *b*. On reaching the score of 18 the link between *b* and *c* is included. The algorithm as it stands above suggests that *c* should be allocated the same UPID as *b*. But this would give it the same UPID as *a*, which would go against the rule that unlinked records should have different UPIDs.

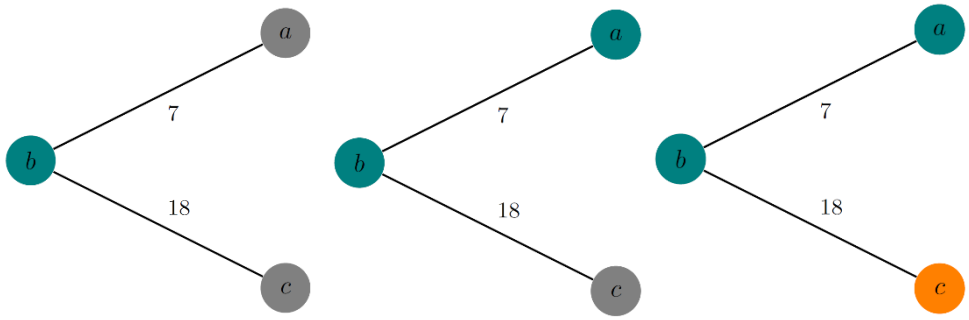

Figure 10 Demonstration of the UPID allocation to a component following the initial allocation. The records are represented by vertices and their links are represented by edges. Letters indicate the record ID and the numbers indicate the distance scores of the links. Colours represent the UPID allocated to each record (grey represents records that have not been allocated a UPID in the initial allocation). The graph on the left shows the situation following initial allocation. The graphs to the right show how the remaining records have UPIDs allocated.

To address this problem the following rule is applied. If there is some vertex k that has the same label as j but is not linked to i at all, then j should be allocated a different UPID from the one that i has. In the example in Figure 10, when the link between *b* and *c* is added then when *c* was vertex i, and *b* was vertex j, then *c* would normally be assigned the same UPID as *b*. However, as there is another vertex *a* (k), which has the same label as *b* (j) then *c* should get allocated a different UPID.

To incorporate this exception the following algorithm is used:

- Loop through vertices k
    - If there is no link between i and k (of any distance score), and there is a link between j and k, then:
        - Do not label i with the UPID of j

The record i may find another record that it could share a UPID with. As before, any record that remains without a UPID allocated will be allocated a different one at the end of the loop.

Exception 2

The second exception concerns records within a particular dataset that are linked. The original IDs from the relevant dataset will help inform whether these records represent the same person. If the IDs are the same then it is likely that they represent the same person, and so there would not be an issue in assigning these records the same UPID.

However, if the records have different original IDs then this may suggest that the records do not represent the same person. It is quite possible that a person might have multiple records on a particular dataset. For example, someone might appear multiple times as a mother on the births dataset, if they have more than one child. However, if there is a situation where a component has multiple records in more than one dataset, then this could indicate a problem (perhaps a pair of twins living together have been linked), and indicate that records for more than one person have been linked.

To address this, before allocating a UPID, a second check is made if record i and j are in the same dataset and have different original IDs. If there is a path in the component from record i to record j that:

- does not use the link directly between i and j, AND
- includes another link between records (other than i and j) that are in the same dataset and have different IDs

then i is not assigned the same UPID as j.

Figure 11 shows two relevant examples. Records left of the dashed line are in dataset *N* and records on the right are in dataset *E*, and all have different original IDs. In the first component the records in question are *N*1 and *N*2. There is another path from *N*1 to *N*2 (other than the link between them), which goes through *E*2. However there is no such path that also includes a link between records in the same dataset with different IDs. Therefore, the exception does not come into effect and *N*1 and *N*2 can take the same UPID. This might involve cases such as mothers appearing on multiple birth registration records.

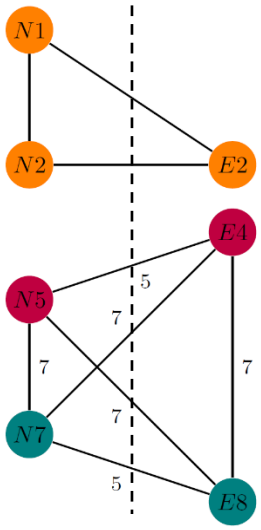

Figure 11 Demonstration of the UPID allocation for two components. The records are represented by vertices and their links are represented by edges. Letters indicate the record ID and the numbers indicate the distance scores of the links. Colours represent the UPID allocated to each record. The vertices to the left of the dashed line are from one source dataset (N), and those on the right are from a second source dataset (E).

In the second component the records of interest are *N*5 and *N*7. Here there are also other paths between these records, but this time there is a path that includes a link between records in the same dataset with different original IDs (e.g. *N*5–*E*4–*E*8–*N*7). In this case there are multiple datasets with apparent duplicates, suggesting that these are indeed different people (perhaps twins), and so they should get different UPIDs.

Another exception covers examples such as that shown in Figure 12. In the example there are two records in the same dataset (*N*5 and *N*7, with different original IDs). This time there is no other path that uses a link between records in the same dataset (with different original IDs). As *E*4 and *E*8 are not linked, they will get different UPIDs. When looping through the distance scores, *N*7 will be the next to get linked, so it will be allocated the same UPID as *E*8. *N*5 will appear at score of 13 when the link between it and *E*8 appears. This would mean that *N*5 would get the same UPID as *E*8. However, this is probably not the desired result, because this means that, although the records in the component are assigned to two different persons, both of the records from dataset *N* are assigned to the same person. Also, record *E*4 is the only record with that UPID. It would, therefore, be preferable for *N*5 to be assigned the same UPID as *E*4. This would mean that the two UPIDs each have two records (from two datasets) assigned to it.

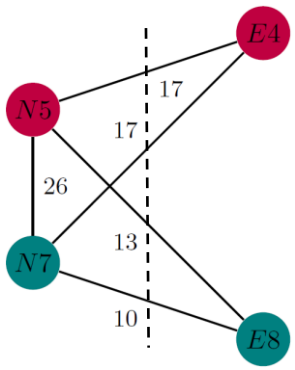

Figure 12 Demonstration of the UPID allocation for a component. The records are represented by vertices and their links are represented by edges. Letters indicate the record ID and the numbers indicate the distance scores of the links. Colours represent the UPID allocated to each record. The vertices to the left of the dashed line are from one source dataset (N), and those on the right are from a second source dataset (E).

Notes on statistical publications

Statistical Research

This publication presents statistical research and the methodology is still under development. We welcome any feedback from users on ways in which the methodology or data sources may be developed to improve the quality of these statistics in future years.

National Records of Scotland

We, the National Records of Scotland, are a non-ministerial department of the devolved Scottish Administration. Our aim is to provide relevant and reliable information, analysis and advice that meets the needs of government, business and the people of Scotland. We do this as follows:

Preserving the past – We look after Scotland's national archives so that they are available for current and future generations, and we make available important information for family history.

Recording the present – At our network of local offices, we register births, marriages, civil partnerships, deaths, divorces and adoptions in Scotland.

Informing the future – We are responsible for the Census of Population in Scotland which we use, with other sources of information, to produce statistics on the population and households.

You can get other detailed statistics that we have produced from the Statistics section of our website. Scottish Census statistics are available on the Scotland's Census website.

We also provide information about future publications on our website. If you would like us to tell you about future statistical publications, you can register your interest on the Scottish Government ScotStat website.

You can also follow us on twitter @NatRecordsScot

Enquiries and suggestions

Please get in touch if you need any further information, or have any suggestions for improvement.

Lead Statistician: Lindsay Bennison

Statistics Customer Services telephone: (0131) 314 4299

E-mail: statisticscustomerservices@nrscotland.gov.uk

For media enquiries, please contact: scotlandscensus@nrscotland.gov.uk