Quality Control and Quality Assurance Strategy for 2021 Census to CCS Person and Household Matching.

Rachel Shipsey & David Edwards, May 2021

Contents

Contained within this paper, as listed in the below contents, are the official paper and 5 other papers, listed as annexes, that contributed to the official paper. The reason for their inclusion as annexes is due to them being internally written papers, that will provide useful insight into the results described in the official paper.

Quality Control and Quality Assurance Strategy for 2021 Census to CCS Person and Household Matching.

Rachel Shipsey & David Edwards, May 2021

Introduction

Matching the people and the households from the census coverage survey (CCS) to the census is an integral part of the population estimation process. Methods used for matching are discussed in **(Shipsey 2021) [Annex C]**. Since the estimation process does not deal well with errors in matching, there are stringent accuracy targets, namely:

- Fewer than 0.1% false positives (incorrect matches) i.e. precision is greater than 99.9%
- Fewer than 0.25% false negatives (missed matches) i.e. recall is greater than 99.75%

To put this into context, in 2011 this meant that there should have been fewer than 650 incorrect matches and fewer than 1,629 missed matches.

In 2021 matches will be made using a number of different methods in sequential order as follows:

- Automatic deterministic matchkeys
- Automatic probabilistic (scoring above threshold)
- Clerical review of probabilistic clerical resolution zone
- Clerical review of matched households containing unmatched persons
- Clerical review of unmatched households containing matched persons
- Clerical review of automatically generated lists of 'best possible' matches (presearch)

Any matched pair of records could potentially be a false positive match. We will use a sampling method to determine an estimate for the false positive rate for each of the different matching methods and then calculate an estimate for the global false positive rate.

Whilst a match could be missed at any of the matching stages, a missed match does not count as a false negative until the final stage of matching – called presearch. This is because at all previous stages, an unmatched record will continue onto the next stage of matching and will still have a chance of being matched. However, the presearch stage is the last stage of person matching – at this stage the clerical matcher will compare the target CCS record with the provided list of census records and either match the CCS record to one of the census records, or declare the CCS record to be unmatchable. If the CCS record is declared unmatchable, but there was in fact a matching census record, then the CCS record is a false negative. This could happen either because the matching census record was never displayed in the presearch list, or the matching record was displayed but the clerical matcher failed to make the match. We will take a sample of CCS records unmatched after presearch, and then use clerical search to see if a match can be found in the census data. This will enable us to estimate the false negative rate.

In addition to the quality assurance described above, we will carry out quality control measures throughout the matching period to ensure that our algorithms and clerical matchers are working as expected and correct any errors as necessary.

This paper describes the steps we are taking to both quality control and quality assure the census to CCS matching in 2021.

Quality Control

The census matching can be split into two periods:

- Tuning period (June – August) when we have some but not all of the 2021 Census and CCS data available
- Live matching period (September – October) when we have all the 2021 Census and CCS data

As the name suggests, during the tuning period we will tune our automatic matching algorithms by taking samples of automatic matchkey matches and clerically reviewing them for errors. Sample sizes will be determined dynamically, initially being dependent primarily on practical factors such as availability of data and clerical resource. This is necessary because the algorithms have been built using 2011 data, and since the data collection method is different in 2021 with more online response, we will need to confirm that the algorithms are working as expected. Note however that we expect the data to be of better quality than the 2011 data which contains many scanning errors. We are aiming for a false positive rate in automatic matching of less than 0.01%.

We expect to get a new delivery of data each week during the tuning period. Each week we will run the probabilistic algorithm and use active learning to select the hardest to classify as matching/non-matching pairs to be reviewed and classified clerically. As more of these *hard* pairs are classified, we expect the parameters of the probabilistic algorithm to converge until no more training data is required.

We will also use the tuning period to find the upper and lower thresholds for the probabilistic matching algorithm. This will be done by taking samples of pairs spread throughout the range of scores and clerically classifying each pair as a match or a non-match. The upper-threshold will be set conservatively so that we are certain that all pairs scoring above the upper-threshold are matching pairs.

Thus, by the time we start the live matching period we will have ensured that the automatic algorithms are working as expected.

Most of the matches will be made automatically, however we expect most of the mistakes to be made by the clerical reviewers. This is understandable because clerical matching is not an easy job - it requires great concentration but is also very repetitive. In addition, it is not possible to provide cast-iron rules as to which pairs should be matched – if such rules existed then we would be able to automate the matching. To ensure that the clerical matchers do as good a job as possible, we have completed research that shows how long clerical matchers should work before having a break and how long they should work each day before accuracy starts to decline (see (**Williams, 2020) [Annex E]**). We have also developed a training programme for the clerical matchers that we have trialled during our rehearsals.

We have developed a clerical matching system (CMS) that the clerical matchers will use. This system has been designed to be as ergonomic, intuitive, fast, and accurate as possible. During the live matching, the work of the clerical matchers will be regularly checked by expert matchers. We expect that around 10% of decisions made by clerical matchers will be reviewed by the experts. Incorrect decisions can be rectified, and further training can be given to clerical matchers where necessary. When a clerical matcher is new, more of their decisions will be reviewed until the expert is sure that they are matching to the required accuracy. Clerical matchers can choose to 'send to expert' from any of the matching screens if they are not sure whether to make a match or not.

To further help with quality control, we have introduced an 'undo' button and a 'return to previous' button in the CMS, so that if a clerical matcher accidentally matches a pair when they meant not to (or vice versa) and realises straightaway what they have done (a case of brain/mouse-finger disconnect) then they will be able to go back to that case and change their decision.

Quality Assurance

As well as using quality control methods during the matching, we will also have a quality assurance process that will enable us to estimate the false positive and false negative rates once the matching has been completed.

False Positive Rate

To estimate the false positive rate, we will need to estimate separately the false positive rate for the matches made automatically and via each of the different clerical matching 'journeys' (individual, associative household and people matching, presearch).

For the automatic matching, we will make use of comparison of the two matching methods (deterministic and probabilistic) to help us decide which records to sample. All record pairs will be in one of the boxes $a - f$ in Table 1. Through the quality control measures undertaken during the tuning phase, we can be confident that matches in box a (matched by both deterministic and probabilistic methods) will be true positives. Record pairs in box b (matched deterministically but scoring in between the upper and lower threshold of the probabilistic method) will be automatically accepted and sampled to determine the false positive rate. Pairs in box c (matched deterministically but scoring below the lower threshold of the probabilistic method) will all be clerically reviewed, as will those in box d (above the probabilistic threshold but not matched deterministically).

Record pairs in boxes e and f would not be accepted automatically and so are not part of the estimate of the false positive rate for automatic matches.

		match keys	
		link	**Non-link**
Fellegi-Sunter	Link (scored above upper threshold)	a (TP)	d (review all)
	Clerical resolution zone (CRZ) (scored between upper and lower thresholds)	b (review sample)	e
	Non-link (scored below lower threshold)	c (review all)	f (TN)

Table 1: This two-way table shows the interaction between the deterministic and probabilistic automatic matching and which types of records will be used for the clerical review to determine an estimate for the false positive rate in the automatic matching.

Once the clerical review is complete, the automatic false positive rate can be calculated using Equation 1. Here x_b, x_c and x_d are the number of false positives found in b c and d, respectively, and n_b is the total sampled in b. Note that the FP rate can also be thought of as being equal to $1 - precision$.

Equation 1:

$$Automatic\ FP\ rate = \frac{\frac{b}{n_b}x_b + x_c + x_d}{a+b+c+d}$$

In Equation 1 it displays how a sample is required from the matched records in box b, whereas for boxes c and d we are able to clerically review all of them. This result is determined from the 2011 Gold

Standard (GS) census to CCS matching results. In the case for box b, the GS shows that for records accepted by a matchkey but for probabilistic are within the CRZ, there are too many (in this case ~220,000) for them all to be checked, meaning that a sample of these is required. However, for boxes c and d in Table 1, the results allow for all these records to be reviewed, as there are a total of ~3,500 for both boxes it total. Delving deeper into the results from the GS for box b , we see that the matches are spread across this CRZ range, meaning that a sample that accurately represents all the possible scores is required. In this instance, the upper and lower bounds of the CRZ are 27 and 15.5, respectively. To generate this accurate sample, it is important to understand the distribution of the matches in the CRZ by looking at their match scores. Doing this, it is noticeable that all the matches lie close to one of the bounds, with the vast majority lying by the upper bound, and very few being in the middle of the CRZ. As the majority that needed to be sampled from had a relatively high score, at approximately 205,000 of the 220,000, it is plausible to adopt a stratification by match score, using optimal allocation.

Optimal allocation can be used here in particular because matches with different match scores are going to have different FP rates, meaning differing variances. This means that within strata variances are small (homogeneity) and between strata variances here are different, justifying optimal allocation. Furthermore, the FP rate for strata close to the upper bound is expected to be very small, which leads to issues generating a small coefficient of variation as the sample size would increase exponentially with very little reward or benefit. Nevertheless, using the GS data generated a suitable sample size of around 20,300, in addition to the records in boxes c and d, totalling around 23,500, and this gives a coefficient of variation (CoV) of around 0.375. If we were to decrease this CoV to 0.3, the sample would increase from 23,500 to over 35,000. Attempting to make the CoV smaller than 0.3 leads to the sample size being implausibly large.

In addition, we will sample records matched using the clerical matching system, stratified by matching journey (individual, associate household, presearch) and using prior estimates of the precision for each matching journey from our rehearsals. Although the matches will have been made via different journeys in the CMS, they will all be reviewed using the individual journey rather than via their original journey. Clerical matchers have access to all additional information, household information and images of the questionnaire (if the response was on paper) via the individual matching journey, so there should be no bias caused by using this journey for all of the review and it is far easier to implement and quicker to carry out than repeating household and presearch matching. We expect to review a sample of less than 4,000 (~4.7%) of the person records that have been clerically matched. This is to obtain an estimate for the false positive rate with a coefficient of variation of less than 0.2.

Lastly, we will sample matched household (HH) records to estimate the false positive rate in the household matching. Household matches can be made as follows:

- complete household matches made automatically with no clerical sight – this occurs if both the household and all the people within the households are matched automatically.
- Incomplete household matches – the households are matched automatically but there is at least one unmatched person – these cases are sent for clerical review.
- Associative household matching – the households are not matched automatically but there is at least one matched person between the households – these cases are sent for clerical review.

Each type of household match has a different level of accuracy. In the case of the latter two, the rehearsal run from April 2021 produced different amounts of FPs when a sample of the records was taken. However, for the complete household matches, we know a household automatic match must

have the same postcode, and all the people must match too. Using this, there is no evidence to suggest a match is wrong unless the person matches are wrong. At this stage we will have checked to verify person matches, therefore, it is determined the complete household matches are 100% correct. In any case, a small sample will be produced to ensure the validity of the assumption. Consequently, when performing the QA, we need only sample from the latter two: the incomplete households and the associative households. As previously mentioned, the rehearsal run through produced FPs for both scenarios, with the rates being 0.00125 and 0.02 for the incomplete and associative matches, respectively. This makes using optimal allocation, when allocating a sample of these records, plausible because of the differing FP rate. From the estimates for the numbers of matches for each scenario from the rehearsal, a sample of size 6,000 is appropriate for determining the FP rate in household matching, with a coefficient of variation of 0.2.

Further details of the sampling strategy for assessing the false positive rate in Census to CCS linkage can be found in **(Edwards, 2021)[Annex A]**

False Negative Rate

As mentioned in the introduction, false negatives can only be made during the presearch stage of matching[1]. Therefore, we only need to take a sample of CCS records unmatched after presearch to estimate the false negative rate. We will then use clerical search to try and find a match for these records. Note that clerical search is different from pre-search because the clerical matcher is able to search the entire census database to try and find a match for the CCS record rather than being restricted to the list of best possible matches that are presented at presearch.

It is very important that the presearch algorithm works well, with the correct match, if it exists, always (or nearly always) being presented in the list of possible matches. To ensure this, when we first start using presearch we will take a sample of 800 records unmatched after presearch and use clerical search to see if any of them are false negatives. Any false negatives will have their match status corrected and we will tune the presearch algorithms/retrain the clerical matchers making use of the reasons for the false negatives to do so. Any records still unmatched will then be put through the revised presearch algorithm and another sample of 800 unmatched records will be taken and reviewed using clerical search to see how many are false negatives. This process will be repeated until we are sure that presearch is working correctly. Based on 2011 data, this would mean that we would stop the cycle of iterative improvement if we observe 18 or fewer false negatives in the sample of 800 unmatched CCS records.

Once the matching is completed i.e., all CCS records have been declared as a match or as unmatchable, we will take a final sample of unmatched CCS records and use clerical search to determine the number of false negatives. This will give us an estimate for the false negative rate. This final sample needs to be large enough to ensure that we are confident that are true value of the false negative rate is less than the maximum allowed false negative rate. However, we also need to keep this sample small

[1] A record may be classified as unmatchable prior to presearch because it has been flagged as 'an issue'. This could occur automatically because the record meets some pre-determined criteria e.g. a student at their out-of-term-time address which is out of scope for matching. Or a clerical matcher may have reported the record as an issue e.g. they see that this is in fact the record of a pet. All clerically reported *issue* records will be verified by an expert matcher. We are therefore assuming that none of these records are false negatives.

enough that it is feasible to complete in a timely manner (clerical search is by far the slowest matching process).

We will be able to use the false negative rate determined during the quality control stage as a prior belief of the false negative rate e.g., we may believe that the false negative rate for presearch is 18/800 (2.25%). We can then use a Bayesian modelling approach to determine the required sample size so that we would be 95% confident that the true value of the false negative rate is small enough to meet the accuracy target. We expect that the final sample will be of size around 2,000. Note that the false negative rate for presearch can be larger than 0.25% because of the large number of true positive matches made in every other stage. For example, if we are 'allowed' 1,629 false negatives, and 60,000 records are unmatched after presearch, then the presearch false negative rate could be as high as 2.7% but the overall matching will still meet the accuracy target of recall greater than 99.75%.

Further details of the sampling strategy for assessing the false negative rate in census to CCS linkage can be found in **(Shipsey & Spakulova, 2021) [Annex D]**.

False Negatives in Household Matching

For our purposes, the definition of a household match is 'the same space with at least one matched person'. This means that in order to be a false negative household match either:

1. At least one person was matched between the households but the household itself was not matched, or
2. There was no person matched between the households, but there should have been, and then the household itself should have been matched.

Based on estimates for the FN rate in HH matching, as well as an estimate of around 40,000 unmatched records in HH matching, we will take a sample of around 8,000 type 1 household pairs and use clerical review to determine the false negative rate in this sample with a coefficient of variation of 0.2. In addition, for any false negative person matches found, we will review the households of these persons to determine if the household should also have been matched – this will tell us the false negative rate for type 2. In terms of the proportion of the total population for HHs, more is needed in the sample of FNs compared to FPs because of the larger proportion (0.0025 for FNs compared to 0.001 for FPs) meaning there is more variability in the population. Furthermore, the FPs can be split into optimal strata whereas the FNs are not. See **(Edwards, 2021)[Annex A]** for more on allocations of strata in census-CCS matching. Once we have an estimate for both type 1 and type 2 FN rates, they can be combined to give us an overall estimate for the false negative rate in the household matching.

Longitudinal Study

As an additional quality assurance step, the census to CCS match lists will be given to the ONS Longitudinal Study (LS) team. This team can provide an independent evaluation of the census to CCS matching for the approximately 1% of records that are in the LS (those records with one of the four secret LS birthdays). This will take the form of an independent matching (or 'tracing') of census and CCS records to NHS records, thus linking census and CCS data to the LS database. This will provide a cross-reference of the two matched sets, original census to CCS pairs and LS census to CCS via NHS pairs, enabling the LS team to identify any differences in match status or matched records. See **(Rowlatt, 2020) [Annex B]**.

Conclusion

This paper describes the quality control measures that we will take to ensure that both the automatic matching algorithms and the work of the clerical matchers are of the highest standard, thereby ensuring that the accuracy of the matching meets our quality targets of less than 0.1% false positives (precision greater than 99.9%) and less than 0.25% false negatives (recall greater than 99.75%). Once the matching is complete, we will also carry out quality assurance procedures to produce estimates of the false positive and false negative rates with appropriate levels of confidence and accuracy. In addition, the Longitudinal Study team will provide an independent assessment of the matching quality for those records that are part of the longitudinal study.

References

Edwards, D. 2021. *Sampling strategy for assessing the FP rate in census-CCS data linkage.* Internal paper, hard copy available on request. [Annex A]

Rowlatt, S. 2020. *Plan for evaluation 2021 Census Coverage Survey to 2021 Census matching quality using the ONS Longitudinal Study*. Internal paper, hard copy available on request. [Annex B]

Shipsey, R. 2021. *Census to CCS Matching Broad Overview*. Internal paper, hard copy available on request. [Annex C]

Shipsey, R. & Spakulova, I. *2021. Bayesian Approach to Sampling Applied to False Negative Assessment of Census to CCS Matching*. Internal paper, hard copy available on request. [Annex D]

Williams, K. 2020. *All day matching Testing Report – July 2020.* Internal paper, hard copy available on request. [Annex E]

Annex A: Edwards, D. (2021). Sampling strategy for assessing the FP rate in census-CCS data linkage. Internal paper

Executive summary

There is a need to assess the false positive rate for the census to CCS matching to ensure that we have met the accuracy target of less than 0.1% false positives. This means assessing the accuracy of the automatic matching as well as the clerical matching. We expect the false positive rate for the automatic matching to be incredibly small and this presents a problem, namely, how to determine the false positive rate when there are very few false positives without reviewing an unnecessarily large sample of automatic matches. This paper sets out a method for estimating the false positive rate by comparison of the deterministic and probabilistic automatic matching methods. We estimate that a sample of around 23,500 automatic matches will be enough to determine the false positive rate of automatic matches with a 95% confidence, given an initial estimate of precision of 99.99% and coefficient of variation of 0.375.

In addition, we expect to take a sample of under 4,000 individual matches made clerically, and another 6,000 household matches made clerically, these having their respective coefficients of variation of < 0.2 and 0.2.

Introduction

Within data linkage there are two types of error that can occur. These are false links (false positives) and missed links (false negatives). Both can occur for several reasons, and it is important to find them to report on the accuracy of data linkage. In census-CCS matching, it is important to estimate accurately both the false positive rate and the false negative rate, as this matching process is imperative in determining an estimate of the population. In this work, we deal directly with false positives. See **Shipsey & Spakulova (2021) [Annex D]** for details on how we will estimate the false negative rate.

A false positive (FP) is when two records are mistakenly identified as a match, and this can occur at any stage in the matching process. This mistake can cause estimates in population to be lower than it should be, as two different people are incorrectly identified as one. For clarity, false negatives have the opposite effect, as one person is left unmatched as two different people. Once all the data has been linked, a false positive rate can be estimated.

Upon defining a false positive rate for the matching in the data linkage, we determine a value for precision. This value is calculated using the formula $\frac{TP}{TP+FP}$ where TP and FP denote the total number of true positives and false positives, respectively. Depending on the goal of the data linkage, the data linker will aim for some level of precision. For this case, in census-CCS linkage, we wish to use these values in estimating the population using Dual System Estimation (DSE), see **Benton (2015).** Therefore, we require the error values to be incredibly small, and we aim for a precision value of at least 99.9%. This means, that for every 1,000 matches made, there can be at most 1 FP.

The issue is that to determine the exact false positive rate and values for precision, all the matches would have to be reviewed manually. However, in the census-CCS linkage, we anticipate ~650,000 person matches, as well as ~370,000 household matches to be made by the various methods at our disposal (See **Shipsey (2021) [Annex C]** for a description of the different matching stages). Because of this large quantity, it is physically unrealistic to manually review all of them, so we have to input a sampling strategy so that we can quality assure (QA) our TPs and FPs. This paper provides a sampling framework to assess FPs in three different areas of our census-CCS matching, these are:

- **Automatic person matching –** These are the person matches made automatically by our match keys or probabilistic Fellegi Sunter.
- **Clerical person matching –** These are the person matches that were not matched by match keys but required the use of the clerical matching system (CMS) instead. Therefore, they needed human examination to determine its matching validity.
- **Household matching –** These are all the matches made for the households. This includes all the matches made with household match keys, as well as households made using clerical review.

Each matching process will have a large quantity of matches to QA for FPs. Therefore, each method requires a sampling strategy to accurately determine the rate of FPs. The following sections provide detail on the sampling strategies that are considered, as well as the derivation of sample sizes for all three matching processes, which culminates in a total sample size required for all the QA of FPs.

Sampling strategy

There are a several different sampling approaches that could be considered when it comes to the clerical review of both the person and household matching. It is important to determine an optimal sample size, meaning that we do not sample too many, but also, we review enough matches to be sure our estimate of precision is robust. There are two main sampling strategies that could be considered for this, namely, these are:

Simple random sampling (SRS) – This is the simplest of approaches, whereby the user would draw a sample from the population of size n. The size of the sample would depend on the prior knowledge of mean or, in our case, the proportion (of false positives).

Stratified sampling – This is a more technical approach, where the total population is split into homogeneous subpopulations (strata), and simple random samples are taken from each stratum. How the strata are formed will depend on the initial data structure, but it is important that each stratum have data that are both collectively exhaustive and mutually exclusive. In other words, data (or a match) will appear exactly once, and only once, within any stratum.

Furthermore, for stratified sampling, we must decide on how our strata will be allocated, in terms of population size, they could be allocated as either:

Proportional allocation: This is where the sampling fraction is the same in each stratum, or in other words, each sample $n_1, n_2, \dots, n_k$, for the k strata, are the same proportion of their respective strata $N_1, N_2, \dots, N_k$.

Optimal allocation: This allocation type does not include proportionality for the sampling fraction. This time, sample sizes are declared depending on the standard deviation of the distribution of variables. Whereby strata with a larger variance will require larger samples and vice versa.

The purpose of using optimal allocation is to optimise the sampling strategy and size based on the variance within and between strata. Whereby when the variance between the strata differs greatly, then it becomes prudent to implement a form of optimality, so each homogeneous stratum is fairly represented in the sample, depending on its variance. Therefore, optimal allocation is only ever optimal when there is variance between groups. So, in context with our census-CCS data, to use an optimal allocation we would require strata in our matched data that have differing variance between groups of matched data. We will see how later how using optimal allocation is more beneficial.

Assessing the FPs in deterministic match key matching using probabilistic Fellegi-Sunter

An idea put forward suggests using both our deterministic (match key) and probabilistic, (**Fellegi & Sunter, 1969),** approaches as a way of identifying false positives in deterministic matching and ensuring the quality of the automatic matching process. Firstly, we know each record pair can have any one of four outcomes:

- A true matching pair, or true positive (TP)
- A true non-matching pair, or true negative (TN)
- A false matching pair, or false positive (FP)
- A false non-matching pair, or false negative (FN)

The two different matching methods (match-key and Fellegi-Sunter) might produce the same or a different outcome for each record pair. This idea can be represented by a two-way table as shown in Table 1.

In Table 1, only boxes a to d will account for FPs. Box a includes the matches that would be matched by a match key and would be above the upper threshold for the probabilistic approach. Knowing this, we would be confident that any match with this attribute could be automatically accepted, and so left out of clerical review. Box c will include matches made by a match key, however, the probabilistic approach determines them to not be matches, as they scored below its lower threshold. Records being described in this way will be much more uncommon than box a, so, it is reasonable to suggest that all of these should be clerically reviewed. Box b includes the remaining records that have been matched by a match key, but for Fellegi-Sunter are between the upper and lower threshold, defined as being in the clerical resolution zone (CRZ). These are records we will be unsure about, but unlike box c, we expect there to be a high quantity of these, meaning that these matches will have to be sampled. Box d includes the only records that are accepted by Fellegi-Sunter but are not by a match key. These also constitute matches that could have FPs, and like c, there will not be many of these, so all of them can be reviewed. Boxes $e$and f include links that are not accepted by any of the two methods, so they can be disregarded in calculating FPs.

		match keys	
		link	**Non-link**
Fellegi-Sunter	Link (scored above upper threshold)	a (TP)	d (review all)
	CRZ (scored between upper and lower thresholds)	b (review sample)	e (review sample)
	Non-link (scored below lower threshold)	c (review all)	f (TN)

Table 1: This two-way table represents the joining of deterministic (match keys) and probabilistic (Fellegi-Sunter) matches. Boxes a and f represent matches and non-matches, respectively, that are automatically accepted or rejected. Boxes b and e represent the matches and non-matches that will be sampled. Boxes c and d represent the records that will all be clerically reviewed.

Once the clerical review is completed, we can then calculate a global false positive rate using the formula:

Equation 1:

$$Global\ FP\ rate = \frac{\frac{b}{n_b}x_b + x_c + x_d}{a + b + c + d}$$

Where x_b, x_c and x_d are the number of FPs in b c and d, respectively, and n_b are the total sampled in b. If it is necessary, whereby the number of matches in c and d is too large to for them to all be realistically reviewed, they may also be sampled from instead. Therefore, the numerator will need to be adjusted slightly, to, for example, $\frac{c}{n_c}x_c$ instead of just x_c. Here, and throughout, we define the FP rate as $1 - precision$.

Research data

To use this two-way approach, it is important to first establish the appropriate upper and lower thresholds for the probabilistic score. In 2021, this will be achieved by clerically reviewing record pairs with varying scores to determine where the upper and lower thresholds should be. For the purpose of this research, we have used 2011 census and CCS data matched on our most recent match key and probabilistic methods. These data have both a probabilistic score and a deterministic match (if it exists). Using these data, we then join the deterministic and probabilistic outcomes and subsequently split into groups that are matched by a match key or not by a match key. The data we have can then further be split into smaller groups by probabilistic score so we can assess the distribution of matches across the score range. Table 2 shows the results of doing this between the scores of 27.5 and 15.5, with grouped scores of 0.5. Note that 27.5, to the nearest .5, is the highest score of any match in these data. The structure from Table 1 can then start to be realised by observing thresholds for the data. Looking at Table 2, it seems that trialling an upper threshold of 27 is sensible. This being because there are very few links above 27 not matched by a match key compared to the 26.5-27.0 range. If 27 is to be the upper threshold, then according to the method, we would automatically accept any match made by a match key which also has a probabilistic match score over 27.0. To check whether we can automatically accept these data points, we can compare all the matches to the 2011 Census to CCS gold standard (GS) dataset. In this comparison, there are matches that are in our research data that have not got a match in the GS. So, in validating this assumption, these are the matches that are checked. In this data there are 306,085 record pairs matched by both methods, and only ~4,000 were not matched by the 2011 GS. A sample of 500 of these was taken for clerical review and there were no FPs, therefore the assumption of automatically accepting these matches seems valid.

Setting the lower threshold at 15.5, then there are 3,109 records below this value that have been matched by a match key, and there are 49,490,693, that have not. From the census data, we expect there to be many millions of data points that are rejected. Having around 3,000 records below this value also makes it a much more sensible amount to clerically review them all. If we increased the lower threshold by 1.5, according to Table 2, we would have an extra 15,000 records to look at which would take up too much time.

Min score	Max score	Matched by match keys	Not matched by match keys
27	27.5	306,085	185
26.5	27	145,446	3173
26.0	26.5	54,424	12,036
25.5	26.0	4,302	10,041
25.0	25.5	360	2,812
24.5	25.0	10	430
24.0	24.5	0	42
23.5	24.0	0	3
23.0	23.5	0	0
22.5	23.0	0	0
22.0	22.5	0	0
21.5	22.0	0	0
21.0	21.5	676	256
20.5	21.0	0	390
20.0	20.5	0	183
19.5	20.0	0	101
19.0	19.5	0	3,077
18.5	19.0	286	2,904
18.0	18.5	102	5,041
17.5	18.0	0	1,356
17.0	17.5	0	440
16.5	17.0	5,716	11,693
16.0	16.5	615	6,155
15.5	16.0	9,493	6,504
-	15.5	3,109	49,490,693

Table 2: This table shows the results of joining the probabilistic and deterministic research data. Each row represents a range of probabilistic score of 0.5. With the columns to the right showing the total number in that score range that have, or have not, been matched with a match key.

You can see how there are many matches that have a match score greater than 15.5, which either need to be clerically reviewed or will be automatically accepted. You can also see the disparity in the number of matches in the bottom row when the match score is low.

Once we have established these threshold values, we can fill in the values in Table 3. For our data we have:

		match keys	
		link	Non-link
Fellegi-Sunter	link	a: 306,085	d: 185
	clerical review	b: 221,430	e: 66,637
	Non-link	c: 3,109	f: 49,490,693

Table 3: An updated version of Table 1 with values coming from the research data. We would automatically accept the 306,085 in box a. Sample from the 221,430 in box b, and clerically review all 3,109 and 185 in boxes c and d, respectively.

So, using the values in Table 3, we accept as TPs 306,085 values, that have been linked using both approaches, and we would clerically review 3,109, which are linked by a match key but not linked by Fellegi-Sunter and 185 that have been linked by Fellegi-Sunter but not by a match key. For the remaining 221,430, we must determine a suitable sample size, this is because we do not want to sample more than necessary, but more importantly, we need to use this sample to generate a final estimated value for precision.

Sampling strategy for the Clerical Resolution Zone (CRZ) matches

We have introduced the different ways in which we can sample the matches that lie in the CRZ, these are using an SRS or stratification. Looking at the distribution of the matches in the CRZ in Table 2, we can see they are disproportionately spread through the CRZ, particularly in the middle of the region where there are very few matches. The problem with using an SRS is that for regions where there is less population density, then these might not be accurately represented in the final example. In our case in Table 2, between match scores 17 and 25, there are hardly any matches, and so in doing an SRS, matches in these score ranges may hardly be represented. This is also the problem looking at the data in Table 4, where the distribution of matches through the CRZ is shown by each match key. In an SRS some match keys may not be fairly represented. Having stratified data will ensure there are homogeneous groups of records in the final sample, and so allowing summaries to be made about them each. It is from this the conclusion to use a stratified sample is made.

Having decided on using stratification of some form, we must decide on how we wish to stratify. In this context, it seems sensible to consider it either by match key or match score as the stratifying method. We know that for homogeneous strata, we must have strata that have similar characteristics, so this initially makes it plausible to consider match key as the strata rule. However, upon observing match keys, we notice that this isn't the preferable case. Table 4 shows how the CRZ is split based on match keys, with some match keys, like match key 8, making up more of the population that others. For those match keys with many thousands included, upon further inspection, we see that the match scores for these matches range through the CRZ. This is an important characteristic as we can say that FPs are more likely for those matches with a lower score, irrespective of the match key it was matched by. Therefore, having match keys as strata does not create homogeneous strata, as each stratum would have larger within group variances. The idea of introducing strata involves decreasing within group variance as best as possible.

However, we can look instead at using match score. In this case, we can say that strata are more homogeneous. This is because we can look at match score as a probability that the pair are a match, with higher scores indicating a greater likelihood of a match. Therefore, FP rates are lower in matches with higher scores, and this remains the case regardless of match key. Because of an expected difference in FP rate based on match score, then the between group FP rates and variances will differ, meaning that optimal allocation for the derived sample sizes is plausible for stratifying by match score.

In summary, the most appropriate method to sampling FPs within the deterministic match key process is to use a stratified sampling strategy, using match score as strata, as well as implementing optimal allocation for the stratum sample sizes. How the scores are split up for the strata will be explained next.

Match key	Number of record pairs in the CRZ		Match key	Number of record pairs in the CRZ
1	140		16	261
2	0		17	15,832
3	0		18	1,168
4	639		19	1,322
5	0		20	9,780
6	19,628		21	4,013
7	3,993		22	1,319
8	91,575		23	9,040
9	5,581		24	814
10	3,268		25	22,312
11	21,698		26	2,892
12	4,288		27	1,237
13	14		28	0
14	132		29	0
15	627		30	0

Table 4: This table shows how the record pairs within the clerical resolution zone (CRZ) from the probabilistic method are distributed between the match keys.

Observing the FPs in the research data

For calculating our sample size and determining the strata, we need to use a prior estimate of the FP rate, which we can obtain using research data and GS data., where the GS data is the standard list of matches of the census to CCS matching from 2011. Firstly, looking at the data organised in Table 2, you can see there are two 'peaks' in quantity of matches at each end of the clerical resolution zone (CRZ) range, with the vast majority being just below the upper threshold. We would expect the 2021 data to follow a similar pattern, and so we can use this knowledge to help with our sample size. It is particularly helpful in determining our prior FP rate estimate, as will be seen shortly.

In determining this estimated prior FP rate, we initially implement a join between the research data and the GS data. Therefore, all the research data, with their respective match score, will possibly have a link to an entity in the GS set, but some will not. In determining how many FPs there are, we look for the records in this join that do not have a corresponding GS link, as these will be the records in the research data that could be a FP.

Initially, we have decided to analyse this data by splitting into three sections, which are shown below in Table 5. For the record pairs with probabilistic score between 15.5 and 17, there are 15,824 in total, (5716+615+9493 using the Table 2 data). Of these, 158 are not within the GS set, and from subsequent clerical review, it produced 7 FPs, so we have, approximately a rate of FPs of 4.5×10^{4}. In the data between 25-27, there is only 1 FP using the same approach out of the approximate 200,000 records. In between 17 and 25, there are not any FPs, but it is quite a small set of just over 1,000 records.

Score range	# in score range	# not in GS	# FPs	FP rate
15.5-17	15,824	158	7	~ $4 \cdot 5 \times 10^{-4}$
17-25	~1,000	11	0	0
25-27	~205,000	2,037	1	~ 5×10^{-6}.

Table 5: Table showing the FPs in the score range. There are not many at all from our data, which is what we expect. We use these FP rates as a basis for producing a suitable sample size. Because these are so small, we are not 100% certain these results will be replicated because the data will be different in the upcoming census, so a slight overestimation may be needed, at the cost of extra clerical work.

We can use these prior estimates for a FP rate to devise a sampling strategy for a fixed target of coefficient of variation. With these values, we can also determine an interval of precision that will accurately reflect our situation, this interval will typically be 'estimated precision $\pm$ 2 standard errors of the estimate'. What you have as a target will depend on the task at hand, but in the census-CCS matching, as seen in the previous paragraph, precision looks to be very high and standard error very small, so our interval of precision will in turn be small. For example, if we have an estimate of precision of 99.95% (0.9995), we could have a standard error target of 2×10^{-4}, giving us an 95% interval between 99.91% (0.9991) and 99.99% (0.9999) precision.

Strata sample sizes

Smith et al (2016) includes formulae for how we can then determine the strata sample sizes for a standard error requirement and a resulting total sample size will come from this. The formula for the stratum size:

Equation 2:

$$n_j = \sum \frac{N_j \sqrt{p_j(1-p_j)}}{N^2} \times \frac{1}{var(\theta)} \times N_j \sqrt{p_j(1-p_j)}$$

Where n_j is the sample size for stratum j, N_j is the size of stratum j, p_j is the precision estimate in stratum j, N is the total population, and $var(\theta)$ is the required variance for the precision.

In the paper there are also several examples for this method, and the first example also demonstrates the effect of optimality of strata, specifically how optimising certain strata will reduce the amount of sampling required. This is something we can replicate in our census-CCS data as the slight change in FP rate through the match score range means we can optimise our sample size too, as one whole SRS will result in pointless extra clerical work.

Following the calculation of a required sample size for each stratum, the variance for each stratum, based on our prior estimates is calculated. It can be calculated using the formula:

Equation 3:

$$stratum\ j\ variance\ = \frac{{N_j}^2 p_j(1-p_j)}{N^2 n_j}$$

We will then have a stratum sample size and individual stratum variance. These stratum variances will sum to obtain the total variance requirement, ignoring potential rounding errors. The total sample size n will be the sum of all n_j. i.e., $\sum n_j = n$.

Once we have calculated our sample and performed the clerical review, we will then have our own estimates of FP rates and its new corresponding variance. It is likely these new values will differ slightly to our prior estimates, meaning that we will have to update our values. This is simply done by inputting our new FP rate p_j into Equation 3 above, so we obtain a new value for variance. We will subsequently have to change our final estimate of precision and interval of precision.

Obtaining a sample size for the CRZ in the census-CCS data

Previously, we have described how the best course of action is to split the data into match score, with the results of Table 5 showing how the FP rate differs, depending on the score. We also have, from Table 5 a small number of matches between the scores of 17 and 25, and looking at Table 2, these are all below 21.5. Table 5 shows how there are no FPs in this range, which could simply be due to the small amount in this area. As these values are below 21.5, there is a gap between 21.5 and the matches at the higher end of the threshold. It seems logical at this point to then split the zone at this point, because of the differing FP rate, into effectively, two new strata. One with matches with a score between 21.5-27, and the other between 15.5-21. This is nearly a direct split of the zone in terms of score, which is useful but also coincidental, meaning it could be different in the 2021 live run.

Therefore, for the CRZ, we now have these two strata. The split made here is reflective of the data we have here to work with, as there are two 'peaks' of matches made at opposing ends of the CRZ. We are also aware that the upper and lower thresholds may be different in 2021, which again is dependent on the data we obtain. Exactly whereabouts the split is made can only be done in due course, but we expect similarities to be present.

So now we have these two main strata and each one will have its own FP rate. This is where we can use the results from Table 5. The middle row will integrate with the top row, giving us the two strata of size ~205,000 and ~17,000. These will be referred to as stratum 1 and stratum 2, respectively. We are also aware that the FP rate is smaller in stratum 1 than in stratum 2, looking at Table 5. This is

where we can introduce optimal allocation because of the differing FP rates. In allocating a sample size for each stratum, we remember we must have an initial estimate of proportion, which should to a degree, reflect what we expect our outcome to be. As we have training data and GS data, we can conjure up a relatively accurate FP rate for each stratum. Relatively is included here as an indicator that of course the training data will not have the same value as the upcoming census, but it should be similar. As a rule, because of uncertainty, it is better to slightly overestimate the FPs we expect, so loosening the restriction on our initial precision estimate. This means we will have to clerically resolve more, but our result will be more accurate, but bearing in mind we do not want to make the clerical work too large.

So, using Table 5, we could have a FP estimate of 5×10^{-6} for stratum 1 and $4 \cdot 5 \times 10^{-4}$ for stratum 2. It is not suboptimal to increase these values slightly, which for this estimate, they are both roughly doubled. Therefore, we have FP rates of 1×10^{-5} for stratum 1 and 1×10^{-3} for stratum 2. An overall coefficient of variation target is also needed to be able to calculate the final interval. In terms of this requirement, it needs to be suitable in terms of the sample size. For example, if the initial standard error estimate is extremely small, then the interval for precision will also be very small. This means that the sample size will have to be large to obtain an estimate that befits this estimate of precision. Once again, a balance must be met in terms of how reasonable an amount of clerical can be done. For the sample sizes below in Table 6, the standard error requirement is $3 \cdot 75 \times 10^{-5}$.

Using these values for our FP rate, as well as using the formulae provided in the previous section, we can calculate the required sample sizes for each stratum. These can then be summed to give us our total sample size, for our specific variance requirement. This is shown in Table 6 below.

Stratum	N_j	p_j=P(false positive)	N_j x p_j	Required sample size
1	~205,000	0.00001	2	11090
2	~17,000	0.001	17	9192
Total	~222,000		19	20282

Table 6: This table shows the CRZ split into its two strata, as well as its corresponding p values and sample sizes. The N_j x p_j column provides the estimated amount of FPs expected in each stratum, and overall.

To calculate the initial estimate of precision we use the total column:

$1 - \frac{p}{N}$ => $1 - \frac{19}{222{,}000} \approx$ 99.99%

As mentioned, these samples are made with a standard error requirement of $3 \cdot 75 \times 10^{-5}$, or a margin of error of $7 \cdot 5 \times 10^{-5}$, (i.e., 2 x standard error.) So, in this instance, the interval would be between 99.9825% and 99.9975%. Moreover, the coefficient of variation would be approximately 0.375.

This sample size is obviously dependent on the standard error we input into the formula. If we reduced our target coefficient of variation, we would have a smaller standard error as a requirement, and we would need to sample more to fulfil this requirement.

As an optional extension, we could further split our data up into substrata. From Table 6, we have our two independent strata, but each one could be split up into smaller groups. The idea behind this is to look more in-depth at the distribution of the FPs depending on match-score. To do this, for each stratum, we order the matches numerically on match score, then split them into groups of even size, say 10,000, for each substratum. There will likely be a remainder at the end, but this is no issue. Because the FP rate will not change between these substrata, we can then take a proportionally allocated sample from each substratum, including the smaller sized one. The total sample size here will be equivalent to the one calculated previously but will just be split through ranges of scores. For example, Table 7 below shows what happens when splitting stratum 1 up into these proportional substrata.

Substratum	N_i	p_i= P(false positive)	Required sample size
1	10,000	0.00001	541
2	10,000	0.00001	541
3	10,000	0.00001	541
…	…	…	…
20	10,000	0.00001	541
21	~5,000	0.00001	271
Total	205,000		11091

Table 7: This table shows how stratum 1 can be split into proportional substratum, in the aim of looking at match score FPs a little closer. You can see for the first 20 each have a size of 10,000 and a sample size of 541, and the final stratum is half the size, so its sample size is half as much, approximately.

The same can also be done to stratum 2, but of course there is not expected to be many there. The most important outcome is that the sample size is the same.

So, whether the strata are split into substrata or not, there will be an outcome from the clerical review of the sample. This outcome will contain potentially different values of FP rate and variance for each stratum, as well as the total population. If our initial estimate is accurate, then we would expect these values to not be too dissimilar. In other words, the estimate proportion should be approximately equal to the sample estimate of the proportion. With these new values, we would update our initial estimate. Therefore, we obtain a new precision estimate for FPs in the CRZ, as well as an interval of precision using the standard error we obtain from the sample.

It is important to remember that the sample size used here is indicative of the training data used, so the final sample size used in 2021 will not be identical. It is very likely to be similar however, whether that be larger or smaller. In any case, the amount of clerical required for the match key FPs in the CRZ will likely linger between 20,000, as a minimum, and 25,000. Decreasing it more so would be detrimental to variance (or alternatively, coefficient of variation).

The global FP rate and precision for automatic matching

The previous section details how the sample size is obtained from the CRZ, which provides a supplementary FP rate and precision estimate for these particular matches. Now, we use this information in combination with the rest of the matches in the automatic matching process, namely the matches in boxes a, c, d from Table 1. These figures are recorded in Table 8 below.

Type of match	Population	Matches reviewed
Matched by both match key and FS (a)	~306,000	0

Matched by match key and within CRZ for FS (b)	~222,000	~20,000
Matched by match key but rejected by FS (c)	~3,000	~3,000
Matched by FS but rejected by match key	~200	~200

Table 8: The matches made for each threshold.

For a, we know that the entire population are TPs, so there is no FP rate, i.e., 100% precision. For b we have a sample recorded of around 20,000 in this case. Whereas for c and d we review all ~3,200 of them. Having reviewed all of these, we will obtain a FP estimate for each group which can be summed together, and then be used to calculate the global precision for all the automated matching. This is done by using Equation 1. Furthermore, we can also acquire the variance for each group listed here and thus the global population variance estimate. Thus, leading to our 95% interval for confidence by square rooting the variance for our standard deviation, and then multiplying by two for the margin of error.

Overall, using this 2011 data, it seems we will use ~23,500 records to come up with a global precision. This being the sum of the ~20,300 sampled and the ~3,200 population from below the lower threshold that are all clerically reviewed. This amount can fluctuate, depending on how we define our thresholds and how much manpower we dedicate to clerical resource. The response rate for the CCS will also contribute to this, which will inevitably change, whether that be increase or decrease, in the 2021 results.

Assessing the FP rate in automatic process - conclusion

In summary, this chapter has introduced a method to QA the automatic matches in census-CCS data linkage. The QA of this data is very important as we want to determine the proper function of our automatic matching and to accurately ascertain the number of FPs in these matches, thus leading to an accurate estimate of the global precision.

The method discussed involves using the probabilistic match score that is assigned to each deterministic match. These deterministic matches are then split into three groups, depending on their match score. The first group includes all the matches that are above an upper threshold, which is a threshold assigned to catch definite matches. The second group are all the matches that are below a lower threshold. The third group will be the remaining matches that are in between these two thresholds.

For the matches above the upper threshold, these are automatically determined as true matches (true positives) without the need to be checked. For the group of matches below the lower threshold, these all must be clerically reviewed. Finally, for the matches between the two thresholds, also known as the clerical resolution zone (CRZ), a sample is obtained from the population. This is because we expect there to be too many for them all to be reviewed.

As the matches in this CRZ have a large score range, it becomes prudent to incorporate an optimal allocation based on this score range. Therefore, the population is split into two strata, based on score range and an appropriate sample is drawn from each stratum. Each sample will then be clerically reviewed.

Once we have completed all our clerical review, we can calculate a global FP rate and precision. Overall, for the deterministic matching, we expect the precision to be slightly greater than our overall precision target of 99.9%, as this final total includes the other, slightly weaker approaches of matching.

From the data used from 2011, it is estimated that around 20,000 need to be sampled overall, in addition to the ~3,000 from below the lower threshold that need to be reviewed. This leaves a total of ~23,500 in the clerical review based on 2011. However, this amount is not set in stone, and can slightly increase or decrease. For example, in the CRZ sample, the sample may change depending on the threshold range to sample from, the total amount to sample from, and requirements needed for the precision, i.e., variance requirements.

In total, this QA method of the deterministic match keys should not need a sample of more than 25,000 as a worst-case scenario.

Assessing the FP rate in clerical resolution.

The previous chapter thoroughly explains the approach for assessing the FP rate in the automatic matching process. However, it is also important that the FPs are assessed in the other methods by which person matches are made. In this upcoming section, we outline the sampling strategy to the matches made in the clerical resolution process.

Initially in our census-CCS linkage, the data are linked through automatic matching processes and this typically deals with 80% of the data. This 80% makes up the links that contain well recorded data. However, there comes a point when the links are made on more illegible records, which means that linkage algorithms cannot adequately deal with these. This is where we introduce the use of the clerical matching system (CMS) to deal with these problematic links. The CMS here used clerical resolution, similarly to how we perform QA on our FPs (or FNs), but it includes the use of several clerical matching journeys to make the initial judgement on whether a link can be declared a match. These are:

- **Individual matching** – Two individual records are compared to one another.
- **Individual associative** – Records are shown as part of their household because either the address or another pair of individuals in the households have already been matched.
- **Pre search** – A single CCS record is displayed with a list of up to 15 census records, which are the best possible matches according to a combination of matching algorithms. A decision is made clerically on whether a match exists within this list.

The advantage of using clerical resolution, rather than algorithms, to make the remaining matches, is that humans have access to more logical reasoning when making informed decisions or assumptions. Especially on more illegible handwriting in census or CCS forms which a clerical matcher has access to. This clerical is only performed on inadequate data, particularly due to time constraints, but it is still a very effective approach for resolving these cases. Despite this, there is the potential for errors, whether they be FPs or FNs. Therefore, like with the QA of deterministic matches, we must analyse the matches made in the clerical matching journey. Once again, this paper only deals with the FPs, see **Shipsey & Spakulova (2021) [Annex D]** for the FNs.

It is expected that a lot of matches will be made in the CMS, so clerically reviewing all of them is infeasible. In the QA of deterministic matches, we had to employ a stratified sample with optimal allocation to deal with this issue, which is like what we can do here. As the links will go through individual matching first, then individual associative and finally pre-search, we expect the better-quality matches to be made in individual matching, and the worst in in pre-search. Therefore, we expect the FP rate to differ between different matching journey, where it is expected to be highest in pre-search, so it is plausible to stratify by matching journey and take an appropriate sample size from each stratum population. Before the live run through of the census-CCS matching in 2021 we conduct

rehearsals, so we have a good idea of what we expect for the FP rate in each matching journey. The values we have are shown below in Table 9.

In Table 9 we also have an estimate for the population size and a FP rate for each stratum (matching journey). We also have an initial estimate for the precision in the CMS matching:

$$Estimated\ precision\ =\ 1-\frac{433}{75709}\ =\ 0.994\ =>\ \sim 99.4\%$$

From these expected FP rates, we can determine a sample size given the population and a requirement for coefficient of variation. It is important to remember that these population figures are unlikely to remain like these estimates for the 2021 live run through. Nevertheless, we have an estimate of precision and validation for stratifying by CMS journey. Using a coefficient of variation requirement of 0.175, we have a sample size requirement as shown in Table 9:

Stratum	N_j	p_j= P(false positive)	$N_j \times p_j$	Required sample size
1. Individual matches	40,513	0.004	162	1,866
2. Individual associative	33,148	0.002	66	1,081
3. Pre search	2,048	0.1	205	449
Total	75,709		433	3,396

Table 9: The required sample size for a coefficient of variation requirement of 0.175.

From this estimated population, false positive values and coefficient of variation requirement, then we obtain a total sample size of less than 4,000, which is an adequate amount that we can sample. Once again, this clerical work is time essential, so it is important that we do not have a sample that is pointlessly too large. We are aided by the fact that we can perform the QA clerical in the same manner as individual matching is performed in the CMS. The advantage of doing this is that the QA will be performed quicker. The alternative would be to QA matches in the same way as they were matched in the CMS. So, for example, the matches made using pre-search in the CMS, would then be quality assured using the pre search tool too. However, the pre search matching takes a lot longer than individual matching or individual associative matching, thus adding a lot of time onto the QA strategy. We aim only to see if a match has been made correctly in the QA process, so the need to add on this extra work here is unnecessary.

Assessing the FPs in the Clerical matching - conclusion

To conclude, the QA of clerical matches involves using a stratified sample based on the 3 different matching journeys in the CMS. These are the individual matching tool, the individual associative matching tool, and finally the pre search tool. Because we expect the level of FPs to be different for each of these methods, as poorer quality data will feed through the CMS tools in this order until it is resolved or declared unmatchable, we can use an optimal allocation for each stratum population to derive the total sample size. Overall, in the QA of clerical, for the figures we have obtained in the lead up to the final run of the 2021 census-CCS data, we expect that the QA sample be no more than 4,000.

Assessing the FPs in household matching

The previous chapters have gone through the method for examining the FPs the person matching processes. In this chapter we discuss the household matching that will also be conducted for the census-CCS data. Household matching is performed mainly to match the households, along with the people inside, that have responded in both surveys; it helps to determine whether people inside a household have been missed or not. Note that the definition of a household match, for our purposes, is that it is the same space and contains at least one matching person.

Although there are not as many as in the individual person matching, there will be many thousands of households to match together. Furthermore, the quality of the response will vary, meaning that some households will be more difficult than others to match. Because of this, FPs can also occur in the household matching, and the frequency of this will also depend on data quality, similarly to person matching. Also, household matching has several stages, where errors in matching can be made throughout.

The three main areas of household matching can be referred to as:

- **Complete household matches –** These are the matches that are made automatically via match keys. This means that the address, and all the people in the households have been matched.
- **Incomplete household matches –** These are the household links that have been matched, yet there are some people within either of the households that have yet to be matched.
- **Associative household matching -** These are the remaining unmatched households, that contain at least one matched person, so there is some association matching required through using clerical resolution. These will be less common, and likely be the most poorly recorded data.

In terms of QA of our household matches, we do not expect to have to clerically review our automatic matches, i.e. we expect 100% accuracy. This is because we will have a common postcode, which is the lowest level of geography estimation will work with, and we will have all persons matched together. All of these cases automatically constitute a match. Nevertheless, we will examine a small number of these matches in the official run through to be certain of this assumption.

For the incomplete household matches, we expect these to be very accurate as well in terms of matching at least one person and agreeing on postcode. From our April 2021 rehearsal conducted on 800 of these types of records, only 1 is determined to be a FP. For the associative household matches, these are expected to be less accurate. This is indicated in the same rehearsal, where the examination of 400 records resulted in 8 FPs.

It is important to note thar both these values for FPs have been trimmed down significantly because of an initial scepticism on how we would define the household match, in terms of the address. This caused doubt on matching households if the address was slightly different, i.e., 33A to 33, even with some identical people, may not be considered a match. However, upon realising that estimation deal only with postcode as the lowest level, the doubt around slight address differences is removed if they have the same postcode. These rehearsal FPs are the result of matches made under this previous doubt, so it could well be these FP rates are overestimates. Nevertheless, we can use these values as a benchmark estimation for FP rate when it comes to QA.

From the research and previous census-CCS data, we have gathered approximations for the matches made in each of the three methods. These are shown in Table 10 below:

Stage	Automatic matches	Pairs sent to clerical	p_j= P(false positive)
1. Complete household matches (u)	~330,000	0	0
2. Incomplete household matches (y)	0	~80,000	0.00125
3. Associative household (z)	0	~4,000	0.02

Table 10: This table shows how the records will be resolved for the household matching. For the complete household matches, all the matches will be automatic. Whereas, for the incomplete and associative household matches, they will be made through clerical.

Note that in Table 10, the estimates for stage 2 and 3 show the estimated number of pairs sent to clerical. Not all the pairs will be matched so the amounts that are sampled from will be smaller than the population size listed here.

Once again, we must sample the matches made from each stage as it is infeasible to review them all. This sample will allow us to estimate the FP rate for both these stages, leading to an overall estimate of precision for the household matching. Stratified sampling also comes in useful for this, as we can use each stage as a form of stratification, because it is expected that both stages differ in terms of expected FPs. The final FP rate will involve both estimated FP rates from each stratum sample, divided by all the household matches, as shown in the table below:

Equation 4:

$$Global\ Household\ FP\ rate = \frac{\frac{u}{n_u}x_u + \frac{y}{n_y}x_y + \frac{z}{n_z}x_z}{u + y + z}$$

where y is the population in stage 2, n_y is the amount sampled in stage 2 and x_y is the number of FPs in the sample for stage 2. The same applies for stages 1 and 3 with the denotations u and z, respectively. The denominator includes the sum of all three process. Precision can then be calculated accordingly as 1 - global FP rate. As we expect there to be no FPs in the automatic process, the need for u in this equation may be negated.

So, we have the estimates for the FP rate in both the incomplete households and the associative households. We can use the previous stratified sampling idea to generate an appropriate sample size for these as well, noting that it is only required on stages 2 and 3. Table 10 shows us how many pairs are sent to clerical, not how many are matched together, so using these values in the formula provided for calculating a sample size would prove to potentially be an over approximation of the matches made. Nevertheless, using it as an upper bound, we would expect to sample between 5-6,000 of these, with around 5,000 coming from stage 2. This would give us a coefficient of variation of around 0.2. But, as mentioned, the pool of matches to sample from should be far less than the ~84,000 given in the table.

Conclusion

To conclude, we have provided a sampling strategy and estimates of sample sizes for the QA of all our possible FPs for both person and household matching. For the deterministic match key QA, this is ~23,500. For the CMS person matching, the requirement is around 4,000. For the household matching,

this number is expected to be around the 6,000 mark. Overall, the rehearsal suggests that for all these methods that ~33,500 records will be required for all the QA. Naturally, we must consider that the responses for the census and CCS will be different than the data we have used here from 2011 and more recent rehearsals. Furthermore, the use of an online response form for the census may make a large majority of the responses free of scanning errors upon processing. The CCS responses will still be obtained by field operatives. External factors like these will alter how much sampling will be required, but internal decisions can cause changes as well including requirements for coefficient of variation Furthermore, if we go back to the deterministic person matching sampling, we have identified two strata with optimally allocated samples. Yet it is possible that the final census data calls for more strata because of the way it is distributed. Nevertheless, each chapter provides the framework for how these specific populations should be sampled, with their own respective, but not exact, sample sizes. Consequentially, we are open to changes in the final sample size for any or all the QA processes, but keeping in mind time constraints that would not allow for drastically more clerical review.

References

Benton, P. (2015) Trout, Catfish and Roach The beginner's guide to census population estimates available from https://fliphtml5.com/wihc/qghm/basic (accessed 5/1/2021)

Fellegi, I. Sunter, J., (1969), A Theory for Record Linkage, Journal of the American Statistical Association, Volume 64, Issue 328

Shipsey, R. Spakulova, I., (2021), Bayesian Approach to Sampling Applied to False Negative Assessment of Census to CCS Matching [Annex D]

Smith, P. et al., (2016), Sampling procedures for Assessing Accuracy of Record Linkage
(Heasman D., Sampling a matching project to establish the linking quality – internal prerequisite paper)

Annex B: Rowlatt, S. (2021). Plan for evaluating 2021 Census Coverage Survey to 2021 Census matching quality using the ONS Longitudinal Study.

Overview

Linkage of the 2021 Census enumerations to 2021 Census Coverage Survey (CCS) responses is required to assess the coverage of the 2021 Census. Evaluation of the quality of this linkage is important to understand where linkage error may occur, as this could impact the quality of a census coverage adjustment.

The ONS Longitudinal Study (LS) can provide an evaluation of the quality of the census to CCS matching, as an independent matching (or 'tracing') of census and CCS records to NHS records can be carried out to link census data to the LS database. This would be in addition to a 10% double-blind exercise planned by Methodology.

This independent LS tracing exercise can provide quality assurance for two purposes:

- A cross-reference of the two matched sets to identify any large differences in match status or matched records;
- A contribution towards a census to CCS matching evaluation.

This document outlines the detail of the proposed process and what analysis will be required.

Linkage Strategy

In previous census years, census enumerations born on one of the four LS birth dates are sent to NHS Digital for 'tracing' to be linked to the Personal Demographic Service (PDS), or similar NHS patient databases. Existing or new LS members are flagged on the NHS system and these tracing results are sent back to the LS team in Titchfield. The LS number is a unique identifier that is shared with NHS Digital; the LS team use this to add the linked census information for those LS members in the existing LS database.

For 2021, census records will be sent in batches as soon as responses are received and processed. There will be no systematic batching, such as grouping by geographical area, as has been the case in the past. NHS Digital have developed a matching strategy based on a combination of automatic, probabilistic and clerical matching methods. Census records will be traced by NHS Digital using the Master Person Service (MPS).

The MPS matching strategy consists of three stages:

1) Alphanumeric trace

As NHS number will not be provided with census records sent for tracing, the alphanumeric tracing is the first stage for census linkage. This relies on exact matching of similar information between the census and PDS and requires family name, date of birth (YYYY / YYYYMM / YYYYMMDD) and gender at a minimum. Other information that may be relevant for census linkage could include other given names, PAF address key and postcode.

2) Algorithmic trace

This stage produces a score of probability that records are a match based on different blocks. This relies on information for family name, given name, gender, date of birth, postcode. This stage uses Soundex to account for differences in spelling.

The four blocks are:

- Block 1: find all matches using family name (soundex), given name (soundex), date of birth (YYYY/YYYYMM/YYYYMMDD)
- Block 2: find all matches using family name (soundex), gender, date of birth (YYYY/YYYYMM/YYYYMMDD), postcode
- Block 3: find all matches using given name (soundex), gender, date of birth (YYYY/YYYYMM/YYYYMMDD), postcode
- Block 4: find all matches using gender, date of birth (YYYY/YYYYMM/YYYYMMDD), postcode. A matching probability score is given for date of birth, gender and postcode separately and the average is calculated to provide a total probability score for each match. If multiple matches arise, the match with the highest probability score is selected as the match.

3) Clerical matching

A team will be recruited at NHS Digital to perform manual (also referred to as operational or clerical) matching for records that have not been linked by the previous two stages. This team will have access to census paper forms to aid their decision making about whether two records should be linked.

More information on the MPS matching strategy can be found in correspondence from NHS Digital.

It is thought to be possible to provide a similar tracing service for CCS records in 2021. Once all of the CCS records are processed, respondents born on LS birth dates will be sent in one single batch to NHS Digital for tracing in a similar way to census records.

The tracing results that NHS Digital will send back for analysis will include: all CCS information sent up to be used in the tracing; the outcome of tracing (traced or not traced, multiple enumeration); and the existing LS number from the NHS system so the LS team can link the CCS information to the LS database and to the census identifier. Being able to link back to a common census identifier is important when

comparing the match status of the census to CCS matching and the CCS to PDS matching. This common census identifier is likely to be the census/CCS resident ID.

It should be noted that new LS members will not be identified during the CCS to PDS matching; only existing LS numbers will be returned. No CCS records will be linked into the LS database for research purposes, as the purpose of this linkage is purely to evaluate the quality of the census to CCS matching.

Requirements

The LS team will receive CCS records once they have all been processed and records with LS birth dates selected. This is expected to be around end of August 2021. The tracing of census records will be the priority so the CCS records will be sent for tracing after the census tracing is complete. This is expected to be from around October 2021. Census to CCS matching will be complete by end of October 2021 and the matched outputs will be sent to the estimation team. Therefore, tracing of CCS records should be complete soon after they are sent to NHS Digital to ensure the value of this evaluation. Although, if the evaluation is delayed until after the Census to CCS matched outputs are provided to the estimation team, this would still be a valuable independent QA.

The expected sample size will be approximately 1% of all CCS records – approximately 6,000 records. This may be less as there was a significant amount of missingness in the CCS date of birth field (~12%) in 2011, which may reoccur in 2021. 1% is a manageable sample size for NHS Digital to trace, and similar in size to annual LS data extracts, but the time and resource required will need to be agreed with NHS Digital. NHS Digital will send the CCS tracing results to the LS team.

The linkage of census to CCS matching will be performed by the Data Integration Team within the Data Access Platform (DAP) environment. The processing of LS data will be done in an Oracle-based system and the analysis of matches will be conducted in a secure environment outside of the DAP (the IL4). Therefore, an interface is required to enable sharing of census and CCS information required for tracing, as well as the matched census to CCS dataset.

The variables required from the CCS extract for tracing can be found in the LS Extract 1 and 2 specifications. These specifications are based on census variables, but it is expected that the equivalent variables will be required for tracing CCS records.

Additional variables that will be required from the CCS extract for this evaluation are listed below. These are needed to enable comparisons of values between the CCS record matched during the NHS Digital tracing and the CCS record matched to the census record during the ONS Census to CCS matching exercise.

Variable	Reason for requirement
Full_DOB_CCS	To identify LS members and compare date of birth during the evaluation.

Born_in_UK (CCS)	To compare information and to aggregate analysis by immigrant status.
Resident_Response_Mode_20 (CCS)	To compare information and to aggregate analysis by response mode.
ETHNIC05_20 (CCS)	To compare information and to aggregate analysis by ethnic group.
ETHNIC19_20 (CCS)	To compare information and to aggregate analysis by ethnic group.

NHS Digital will return the match status, matching method, probabilistic scores for such links and existing LS numbers for each record sent for tracing.

The variables required from the census to CCS matched dataset are:

Variable	Reason for requirement
(Census)Resident_ID	A unique identifier to enable linkage between the census to CCS matched set and CCS to PDS matched set, to compare match status of individual records.
(CCS)Resident_ID	A unique identifier to enable linkage between the census to CCS matched set and CCS to PDS matched set, to compare match status of individual records.
Match_status	To enable comparisons of match status of individual records (0 if no match or 1 if match).
Match_score	To identify the strength of probabilistic matches if there are discrepancies (1 to N as appropriate if match made on matchkey 1-N, or 50 if made by probabilistic , or 99 if made by clerical. Null if Match_Status = 0).
Duplicate_Flag_CCS	To analyse multiple CCS records (1 if 2 or more CCS HHs match to the same CEN HH, Null if Match_Status = 0, otherwise 0).
Duplicate_Flag_CEN	To analyse multiple census records (1 if 2 or more CCS people match to the same CEN person, Null if Match_Status = 0. Otherwise 0).
Cluster_Number	Integer for unique households and identification of multiples. This number will only appear once, for duplicates it will

appear multiple times. Null if Match_Status = 0).

Both duplicate flags and the cluster number are needed to help to sort out cases where e.g. the LS method links one CCS record to one census record and leaves another CCS record unmatched, but the ONS method thinks that the unlinked CCS record is a duplicate. This may help in untangling some of the differences between the two methods.

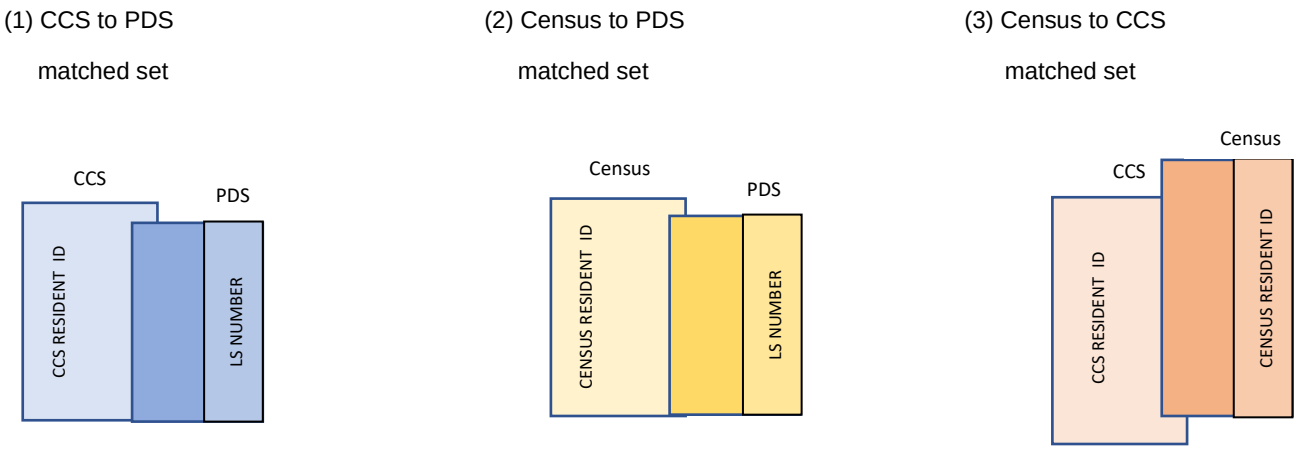
To get extra information for comparing record-level information from the Census to CCS matched dataset, the LS team can link this information from the CCS specification provided before tracing via Resident_ID. This includes CCS information for: sex, age, born in the UK, response mode, ethnicity and CCS postcode (for aggregating to higher geographies.

Different cuts of census data will be taken for the LS census extract and the census data used in the census to CCS matching. The LS census extract is taken before some census cleaning processes, such as Resolve Multiple Responses, so could include records that are not in the census data used in the matching. To overcome this, the LS team will remove any census records from their census data that do not appear in the census extract for matching.

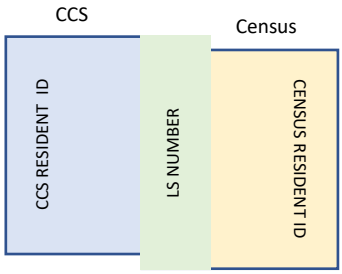
The census to CCS matched dataset will include all census records. Those of interest will only be individuals with LS birth dates and those residing in a CCS area. These will need to be filtered by the LS team before analysis.

The LS team will then need to link the LS number to the matched census to CCS set to enable analysis of records that do or do not match to the same resident ID. This will be done using the method outlined in Figure 1.

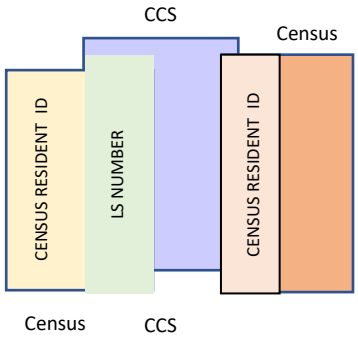
Figure 1: Linkage of LS number to enable resident ID comparison



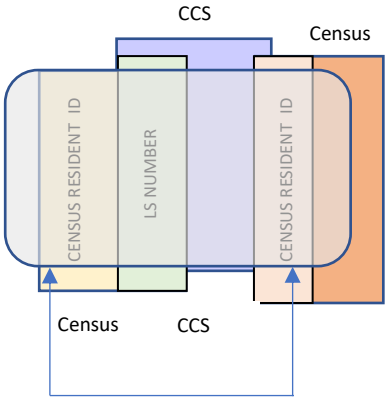
(4) Link (1) to (2) on LS number to create a LS proxy Census to CCS matched set



(5) Link the matched Census Resident ID from (3) to the matched set in (2) on CCS Resident ID



(6) Compare the census resident IDs for records that linked



This linkage will produce a LS Census to CCS matched set by proxy (4), using the LS number. Obtaining a resident ID from the two census-matched sets using the LS number could create some cases that do not match because LS number was missing in the linkages to the PDS. These cannot be compared but should only affect a small number of cases (i.e. the amount of unmatched records).

Steps (5) and (6) of Figure 1 could also be configured to explore where the matched CCS Resident ID from (3) is linked to the matched set in (2) on census Resident ID. This could provide insight into census records for LS members in CCS areas that

didn’t achieve a link, or where the LS tracing finds a link whereas Census may not (or vice versa).

Analysis

The LS birth dates must be protected throughout this work, so only members of the LS team in Titchfield will have access to these and the subsets of data sent and received during tracing. Therefore, the analysis of the quality of the census to CCS matching will be performed by members of the LS team and aggregate analysis only will be provided for the evaluation.

The following aggregate statistics can be produced for CCS to PDS matches for comparison with the census to CCS aggregates (performed by the LS team):

- The number and/or proportion of CCS records traced and not traced
- The number and/or proportion of CCS records identified as multiples

It may be possible to group these analyses by other variables of interest (e.g. age, sex, geography, response type); however, sample sizes may be too small to release so this may not be possible for disclosure reasons.

Record-level comparisons will also be possible and will likely be of most benefit to the evaluation. The resident ID can be used to explore if a match status (i.e. linked or not linked) is different for a record between the census to CCS and CCS to PDS matching; or the matching processes have linked to different records (found by looking at the resident ID – see Figure 1). The LS team could then clerically explore both pairs and use supporting information to make a decision on which is correct and why they are different. This could explore why differences occurred, and could identify clerical error. This will be dependent on timescales.

The record-level analysis will not be provided to Methodology or other teams, as this will identify the resident IDs (and therefore birth dates) of LS members. For the evaluation of the census to CCS matching, aggregate analysis of the following will be provided:

- The number and/or proportion of CCS records that both had a match status of ‘matched’ and matched to the same resident ID
- The number and/or proportion of CCS records that both had a match status of ‘matched’ but matched to a different resident ID
- The number and/or proportion of CCS records that had a match status of ‘matched’ in the census to CCS matching but not the CCS to PDS matching
- The number and/or proportion of CCS records that had a match status of ‘matched’ in the CCS to PDS matching but not the census to CCS matching

Again, it may be possible to group these analyses by other variables of interest if sample sizes allow. This could also be expanded to match type i.e. during which stage the match was made (deterministic, probabilistic or clerical), and match key (if applicable).

The other side of this evaluation can inform the quality of the LS matching process. An evaluation of the following analysis could inform error here:

- The number and/or proportion of existing LS numbers that linked to different resident IDs during the Census to PDS and CCS to PDS matching.
- The number and/or proportion of CCS records that had a match status of 'matched' in the census to CCS matching but not the CCS to PDS matching
- The number and/or proportion of CCS records that had a match status of 'matched' in the CCS to PDS matching but not the census to CCS matching

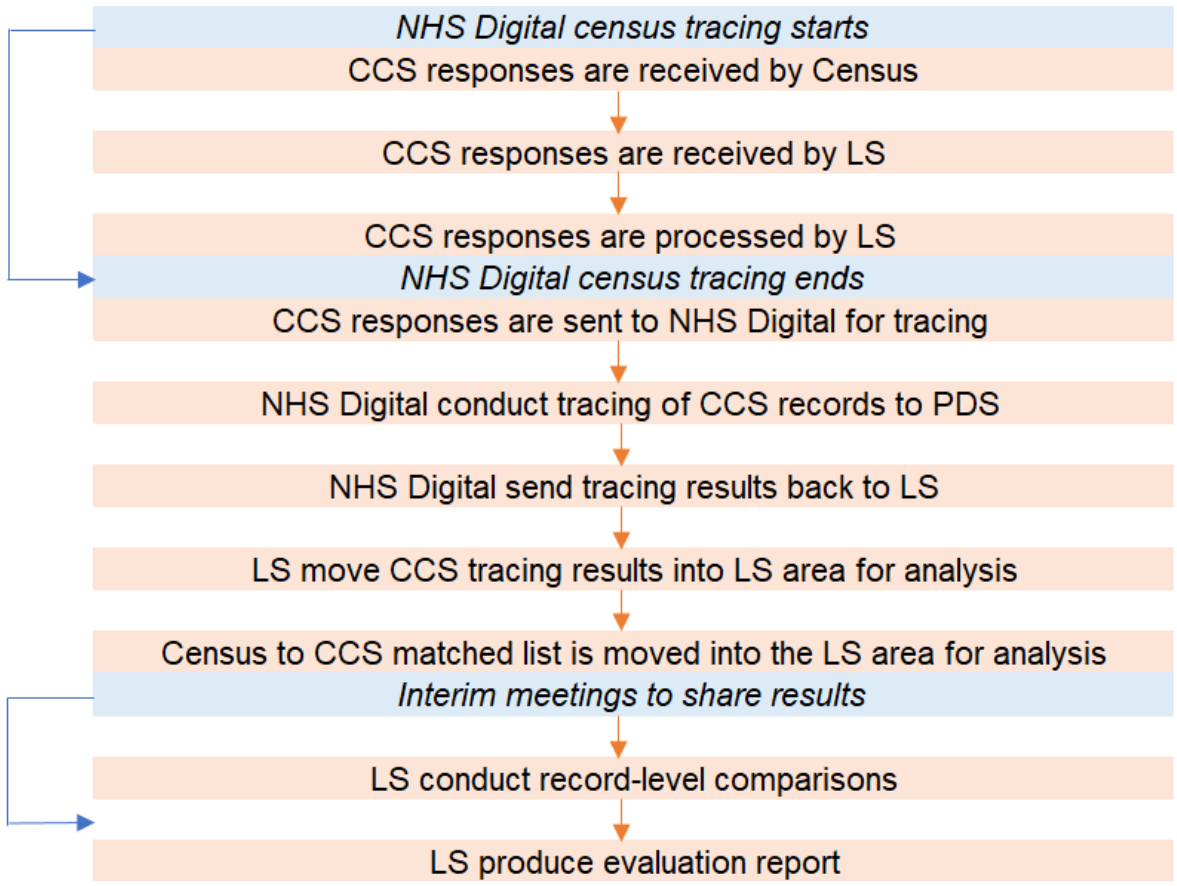
Further clerical exploration could be carried out to identify where the error is most likely.

To communicate the results of this evaluation, interim meetings can be scheduled to provide updates on progress and analyses. A final report including the aggregate analyses and an evidence-based evaluation of why differences may have occurred will be provided to the relevant teams. This can be used to inform a wider assessment of the quality of the census to CCS matching.

Proposed process

The flow chart below shows a simplified process flow of the evaluation.

Figure 2: Process flow for census to CCS matching evaluation using the LS



Risks and dependencies

Risks:

- An interface cannot be set up in time to share CCS data or the matched census to CCS dataset securely. This could mean needing to conduct the analysis in DAP, which will require gaining accesses, setting up projects and upskilling in DAP and DAP-permitted programming languages for the LS team. There will not be enough time to do this alongside all other census-related and business as usual LS work. Discussions should start immediately on how to share data securely from DAP to the Oracle-based LS system.
- The tracing of census data by NHS Digital is delayed and impacts on when they can conduct the tracing of CCS records. This could misalign deadlines of the other planned evaluations and end up not being useful. Regular progress meetings will raise any issues early so contingencies or alternatives can be discussed here.
- There are network/system/software issues within any of the environments (Oracle, IL4, SQL, Stata). This could delay processing or analysis. These are likely to be unforeseen but prior knowledge to any updates etc that may cause these issues would be helpful in advance.
- NHS Digital do not have the resource to complete CCS tracing. This must be included in the contracts agreed for the LS-Census 2021 link project.
- Loss of experienced LS research team staff or recruiting new staff could cause resourcing issues or more time needed to train newer staff and delay analysis. Agreeing plans in advance and understanding what skills are required to train new staff early will help this. Training AIT ROs will also act as a contingency.

Dependencies:

- NHS Digital completing the tracing of census data on time as this is when tracing of CCS records will begin.
- Census providing the CCS records for processing and the census to CCS matched file on time.

Annex C: Shipsey, R. (2021). 2021 Census to CCS Matching Broad Overview.

Introduction

Although every effort is made to count everybody in the England and Wales census, it is inevitable that some people will be missed, and others will be counted more than once. ONS wants to publish an accurate estimate of the population, so we try to work out how many people have been not been counted in the census, and how many have been counted too many times. It is important to note that people who are missed or counted multiple times will not be at random. Carrying out a census is increasingly hard, and the numbers of people and households missed is becoming more significant. Furthermore, the missing people do not occur uniformly across geographical areas or across other sub-groups, such as age and sex groups.

In order to estimate the number of people who have been missed from the census, we carry out a smaller survey called the Census Coverage Survey (CCS) just after the census. We then match the people and the households from the census and the CCS to each other. This tells us how many people completed both the census and the CCS, how many completed the census but not the CCS and vice versa. Using this information, it is possible to estimate how many people completed neither the census nor the CCS using dual system estimation (see [Benton, 2015]).

Any errors in the matching cause the population estimates to be wrong. Put simply, every missed match (known as a false negative) adds to the population estimate because one person is counted twice. Similarly, every incorrect match (known as a false positive) causes the population estimate to go down because there were two distinct people, but we incorrectly thought they were just one person. We want our estimates of the population to be as accurate as possible, so this means that the matching must be as accurate as possible.

To keep our population estimates as accurate as possible, we have stringent accuracy matching targets:

- Less than 0.25% false negatives (missed matches) – this means that we find at least 99.75% of all the possible matches. In other words, we find 9,975 out of every 10,000 possible matches.
- Less than 0.1% false positives (incorrect matches) – this means that at least 99.9% of the matches that we make are correct. In other words, for every 10,000 matches we make, at most 10 of them are incorrect.

In order to achieve this level of accuracy we need to use a mixture of both automatic and clerical matching. This paper describes the different stages of the matching process. Figure 1 shows how the automatic matching and clerical matching stages work together.

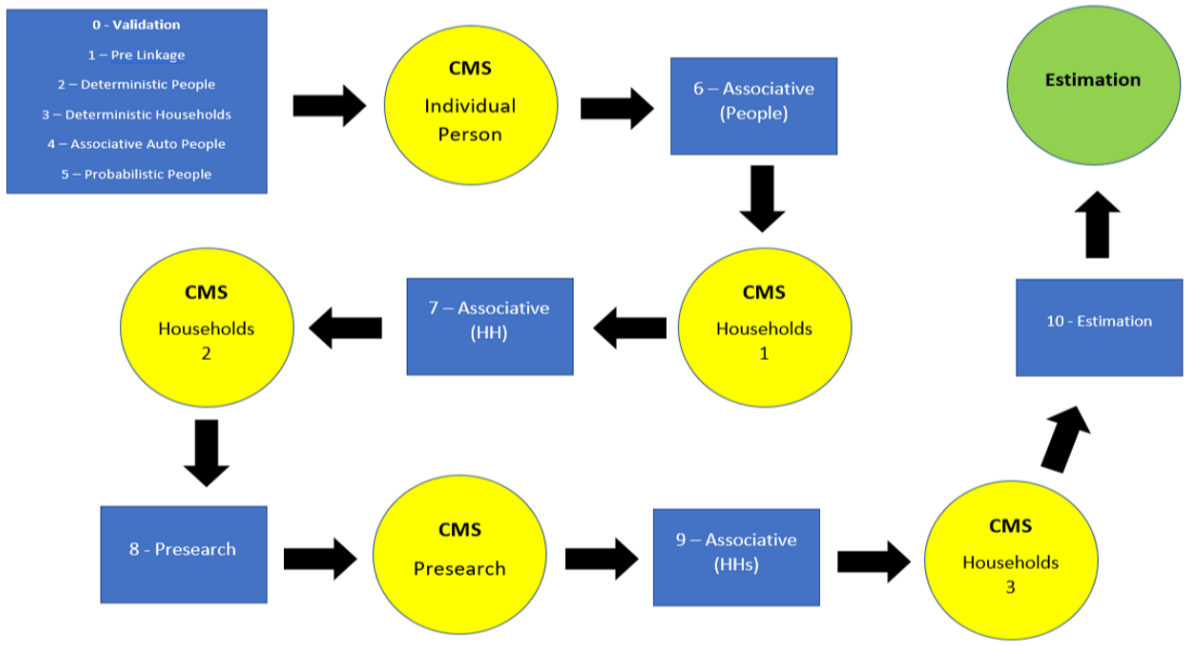


Figure 1. Automatic matching and clerical matching stages of census to ccs matching strategy

Cleaning and standardisation

Before we start matching, we clean and standardise the census and ccs data. This includes the following steps:

- Run diagnostic tests to ensure that all values are in expected ranges.
- Run counts of missingness on the variables used for matching to ensure that there is not unexpectedly large amounts of missing data.
- Replace missing value in names to '' so that string functions can be applied. After cleaning, all missing values are made null.
- Convert all text to uppercase.
- Create derived variables used for matching, including:

    - Fullname – concatenation of all name variables
    - Alphaname – letters of fullname in alphabetical order
    - Nickname – root name of the given forename e.g. forename1 = JIMMY, nickname = JAMES
    - Split forename and surnames that contain spaces or hyphens into separate strings (FN1, FN2, FN3, SN1, SN2)
    - Postcode area, postcode sector and postcode district are derived from postcode
    - Create house and flat number variables from address
- Ensure all variables are saved as the correct type.
- Set aside students at their non-term-time address. These are included in the final outputs as unmatched out-of-scope records.
- For each household create sets of forenames, surnames, dates of birth and ages for use in household matchkeys.
- Mark empty households as out of scope.

Deterministic person matching

We use deterministic, or rule-based, matching in order to find the 'easy' matches. We have designed a set of matchkeys that are used to match people between the census and the CCS. A matchkey is a combination of variables on which two records must agree in order to be declared a match. For example, the first matchkey requires the records to match exactly on full-name (a concatenation of forename, middle name and surname), date of birth, sex and postcode; the eighth matchkey requires a *fuzzy* match on forename and surname, and an exact match on date of birth, sex and postcode.

A fuzzy match on a name means that there is some error, possibly caused by scanning, transposition, phonetic or spelling errors. We use Levenshtein, Soundex and Jaro-Winkler string comparison methods to look for fuzzy matches between names. We also use alphaname (putting all the letters in a name into alphabetical order), common nicknames, and transposition of first and last or first and middle name to make name matches. In addition, we allow for common errors in dates of birth and address information.

There are currently 30 person matchkeys which are listed in Appendix A. The matchkeys are hierarchical with the strictest matchkeys first. We will use the uniqueness of matches made per matchkey in order to determine the correct hierarchical order i.e. a strict matchkey should make very few non-unique matches whereas a looser matchkey is likely to make more non-unique matches. We use a non-greedy approach – this means that even if a match for a record is found on one matchkey, the matched records are still compared on later matchkeys. When all matchkeys have been run, any records that have unique one-to-one matches are accepted automatically, with only the lowest number matchkey being recorded if the same pair is matched on multiple matchkeys. Any records that are part of a non-unique match are sent for clerical resolution (see Clerical matching). This enables us to detect duplicates in the census and ccs data and helps us to get difficult matches correct, e.g. when there are twins living at the same address or father and son with the same name. This is illustrated in figure 2.

Matchkey 30 matches people who are at different locations, but where we can detect that multiple people have moved from one address to another (multiple moves). This indicates that a family has

moved to a new house and is evidence that the person matches are correct despite the mismatch in geography. There is an additional matchkey, number 31, that generates matches using name, sex and date of birth information only (no geography and no evidence of multiple moves). All the pairs generated by matchkey 31 are sent for clerical review.

Note that if the probabilistic algorithm (see Probabilistic person matching) causes a unique matchkey match to become part of a non-unique match, then this cluster of non-unique matches is sent for clerical review.

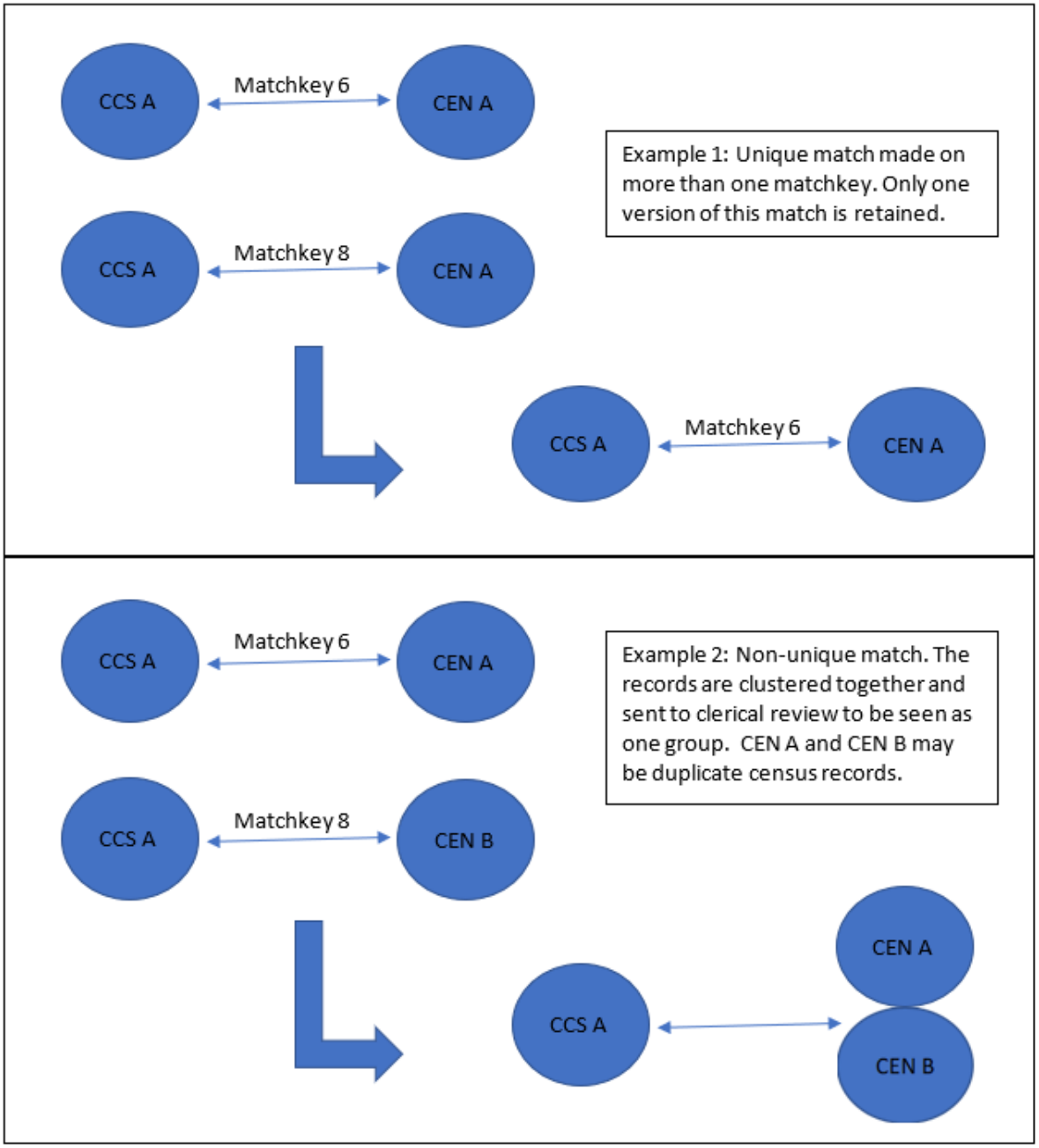


Figure 2: Examples of unique and non-unique matches made by the matchkeys.

Probabilistic person matching

The matchkeys have been designed based on the 2011 Census and CCS data and the 2019 Census and CCS Rehearsal data. This is the best training data that we have, but we expect the 2021 Census and CCS data to be different from 2011 because 2021 is an online first census, and people's attitude to government and data collection has changed. So, the 2021 data might be better (less handwriting and scanning errors) or worse (more missingness) or just different (more typos). The 2019 Rehearsal data is unlikely to be representative of the 2021 responses because it was a voluntary survey. To account for this, we are not relying solely on the matchkeys, but also implementing a Fellegi-Sunter probabilistic method [Fellegi, 1969]. Fellegi-Sunter is a well-known tried and tested probabilistic matching algorithm. It does not require training data but can be optimised using the live 2021 data.

We run the probabilistic algorithm on all the census and CCS records. First, we bring together possible matches by blocking on postcode i.e. bring together as a candidate matching pair every census record and CCS record that have the same postcode (we include the enumeration address postcodes for census and CCS, census day address postcodes given at CCS address and alternative addresses given for census and CCS). This creates millions of candidate pairs. We expect nearly all (98.5% based on 2011 data) of the correct matches to agree on postcode, so we exclude very few correct candidate pairs by blocking only on postcode. The few correct matches that disagree on postcode will be matched at a later stage.

Next, the Fellegi-Sunter algorithm is run and awards each candidate pair a score depending on how well the candidate records matched on forename, surname, date of birth and sex. The score obtained by each candidate pair is used to split the pairs into three groups. Candidate pairs scoring above an upper threshold are accepted automatically. Candidate pairs scoring below a lower threshold are rejected automatically. Candidate pairs scoring in between the lower and upper thresholds (known as the clerical resolution zone) are sent for clerical review.

The Fellegi-Sunter algorithm uses m and u parameters for each of the matching variables. The m-value is the probability that a variable agrees on the census and the CCS given that the candidate pair are a true match. The u-probability is the probability that a variable agrees on the census and the CCS given that the candidate pair are not a true match. The m and u values will be initialised using the matches made by the matchkeys and iteratively adjusted using the new data as matches are made in 2021 by the clerical matchers. We will use active learning [Sarawagi, 2002] to choose which record pairs should be classified as a match or non-match first in order to train the algorithm most effectively. We have found that classifying the most uncertain records first i.e. those in the middle of the clerical resolution zone is most effective.

We expect the pairs that are automatically accepted to mirror the pairs automatically matched by the deterministic matchkeys. If this is not the case, we will investigate the differences and amend the matchkeys accordingly. Using both deterministic and probabilistic methods gives us confidence that our matching methods are working as expected – it is like having a safety net. As well as providing a quality assurance for the matchkeys, the purpose of the probabilistic algorithm is to generate candidate pairs for clerical review.

Deterministic household matching

As well as matching people, we have to match census and CCS households. A household match is defined to be 'the same living space with at least one of the same people'. We have developed a series of household matchkeys which are listed in Appendix B. These matchkeys use household information such as addresses, UPRN, tenure type etc and also person information.

In 2011 we used a 'head of household' variable that was derived on both census and CCS households using an algorithm to pick a person to represent head of household and matching made use of just this person. In 2021, we have a much-improved method whereby we create list of forenames, surnames and birthdates with each household, and look for a match across these sets. Thus, we are no longer reliant on the 'head of household' being present and selected as head of household in both the census and CCS household. So, for example, household matchkey1 requires agreement on UPRN together with a forename, a surname and a date of birth of a person or persons within the household; household matchkey 15 requires the records to match on postcode, tenure, type of property, number of residents, together with a forename and surname of a person or people within the household. This method could mean that we accidentally make a household match where there is no person matching. In practice we have found that this does not happen. However, we will check that all matched household do contain a matched person at the end of the matching process and break any household matches where this is not the case.

As with person matching, any non-unique household matches made at this stage are sent for clerical resolution.

Note that we have deliberately allowed the household matchkeys to be less strict than the person matchkeys. This is because in an incorrect household match, it is very unlikely that all of the people will also be matched. Hence the households will contain at least one unmatched person and will therefore be sent to clerical matching as part of Associative matching. At this point the clerical matcher can break the household match if it is incorrect. Including these less strict matchkeys with the safety net of clerical matching allows us to make more matches automatically and reduces the overall burden for the clerical matchers.

Clerical matching

Although we expect to complete around 85-90% of matching automatically, we still need people (clerical matchers) to look at the harder cases and make the match decision. Some decisions cannot be made automatically because there is too much error in the data, missing data, more than one possible match etc. For example, a clerical matcher can use their human intuition to match the following two records which include a mixture of scanning errors, formatting and naming differences. If we tried to capture this match automatically, we would introduce too many false positive matches.

Census: CATHERINE ELIZABETH WILLIAMS, 10/09/1972, FLAT A 16 GROVE ROAD, PO154NW

CCS: KATY VVILLAIVIS, 09/10/1972, 16 GROVE ROAD GROUND FLOOR APARTMENT, PO154NW

We will employ a team of clerical matchers to look at these harder cases, and we have developed a clerical matching system (CMS) which will enable the clerical matchers to see more information about the person they are trying to match including a view of all the people in the household, extra data from the census form, and the scanned image of any forms that were completed on paper rather than online. Some screen shots showing the CMS are included in Appendix C.

Note that it is possible for a clerical matcher to make a different decision on a record pair than a previous clerical matcher at an earlier stage of clerical matching. For example, suppose clerical matcher 1 sees record pair {census_person_A, ccs_person_B} during individual matching and decides that this pair is a match. If there are still unmatched people in the households containing census_person_A or ccs_person_B then these entire households will be sent back to clerical review at the associative matching stage. Now clerical matcher 2 decides to break the match between census_person_A and ccs_person_B. In these cases of altered decisions, we will always prioritise the most recent clerical decision.

Associative matching

By this stage we will have matched a lot of people and a lot of households. Next, we make use of the matches that we have already made to try and make some more using associative matching. We bring together unmatched people who are contained in households that have been matched. We use the Fellegi-Sunter probabilistic algorithm again to give them a score. If they score above a threshold, which can be a little lower than for the original people matching, then we accept them automatically. All other unmatched people contained in matched households are sent for clerical review. Similarly, we bring together any unmatched households that contain matched people and send these to clerical review. The cleric sees a household view including all the people and any matches that have already been made rather than seeing a cartesian join of all the unmatched people in the household.

Pre-search

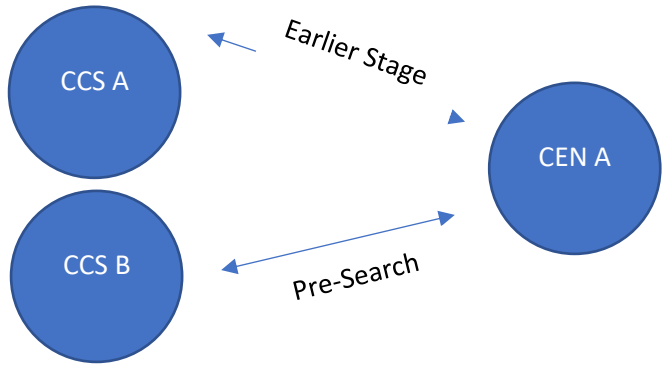
By this stage we have made nearly all the matches. We estimate that there might be around 65,000 unmatched CCS records at this point which include 2,000 CCS records that truly have a match to a census record. Note that these figures are only rough estimates because the true numbers will depend on response rates and quality of the data both of which are unknowns.

In 2011 clerical search (a clerical matcher is given a target CCS record and searches manually in the census dataset to try and find a match) was used to find matches where they exist. This is a very slow process, so for 2021 we have developed the pre-search algorithm that finds the best possible census matches for each unmatched CCS record. The possible matches are presented to a clerical matcher who decides whether any of these are a correct match. If not, then the CCS record is declared as unmatchable i.e. there is no matching census record.

The pre-search algorithm is still under development but is expected to use a variety of methods including matchkeys, probabilistic methods and machine learning with active learning in order to find and rank the best possible matches for each unmatched CCS record.

Pre-search is applied to the remaining unmatched CCS records only. However, all census records are considered as potential matches for these CCS records, regardless of previous match status. This enables us to make duplicate matches where appropriate. For example, census_person_A may have been matched to ccs_person_A in an earlier stage, but census_person_A can also be matched to ccs_person_B in presearch, creating a duplicate/non-unique match. This is illustrated in Figure 3.

Figure 3: Non-unique match created at the presearch stage



Pre-search only applies to person matching. However, we make some additional person matches here so this leads to a further associative matching step where look at the unmatched households containing newly matched people to see if we can make some more household matches.

Quality Assurance

Quality control measures will be taken during the matching process. We will take samples of automatic matches, clerically review them and update the algorithms accordingly, until we are satisfied that the matchkeys and probabilistic algorithms are performing well. Samples of the clerical matching decisions will be reviewed by expert matchers throughout the clerical work and further training offered where appropriate if there are mistakes being made. Quality control ensures that the processes are working as expected or are improved if this is not the case. Matching decisions will be changed if errors are found during the matching.

In addition, there will be a quality assurance process that is used to enable estimation of the false positive and false negative rates. This is carried out after the matching is complete. We will clerically review samples of automatic matches and clerical matches, stratified by matching score and clerical matching journey (individual, household, presearch) to estimate the overall false positive rate. Since presearch is the final stage of matching, false negatives can only come from this stage (false negatives from any previous stage will simply pass to the next stage and if a match exists it can still be found). To estimate the false negative rate of the presearch stage, we will use clerical search (a clerical matcher uses the clerical matching system to search for a matching census record given an unmatched CCS record) to look for a match for a sample of records for which no match was found at the presearch stage.

Outputs

Finally, the results from the automatic and clerical matching are combined to produce a list of matching census and CCS records and unmatched census and unmatched CCS records. Various flags are added to the outputs, for example to indicate whether the matching records are at the same or a different location, whether the matching records are duplicates, or whether an unmatched record is out of scope for some reason (e.g. baby born after census day). The format of the output tables and the list of out of scope reasons as shown in Appendix D.

References

Benton, P. (2015) Trout, Catfish and Roach The beginner's guide to census population estimates available from https://fliphtml5.com/wihc/qghm/basic (accessed 5/1/2021)

Fellegi, I., Sunter, A. (1969) A Theory for Record Linkage. Journal of the American Statistical Association, Vol. 64, No. 328

Sarawagi, S., Bhamidipaty, A. (2002) Interactive deduplication using active learning. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 269-78. https://dl.acm.org/doi/10.1145/775047.775087 (accessed 21/1/2021)

Appendix A

Matchkeys (individual matching)

Note that these matchkeys are correct as of 28/05/2021 but are still undergoing development and are subject to change when we have access to 2021 Census and CCS data.

```
# MK1 FullnameNS, DOB, Sex, PC
    cond01 = [df1.fullname_ns_ccs == df2.fullname_ns_cen,
              df1.dob_ccs == df2.dob_cen,
              df1.sex_ccs == df2.sex_cen,
              df1.pc_ccs == df2.pc_cen]

    # MK2 = FN, SN, DOB, Sex, UPRN
    cond02 = [((df1.fn1_ccs == df2.fn1_cen) | (df1.fn_ccs == df2.fn_cen)),
              ((df1.sn1_ccs == df2.sn1_cen) | (df1.sn_ccs == df2.sn_cen)),
              df1.dob_ccs == df2.dob_cen,
              df1.sex_ccs == df2.sex_cen,
              df1.uprn_ccs == df2.uprn_cen]

    # MK3 = FN, SN, DOB, Sex, PC
    cond03 = [((df1.fn1_ccs == df2.fn1_cen) | (df1.fn_ccs == df2.fn_cen)),
              ((df1.sn1_ccs == df2.sn1_cen) | (df1.sn_ccs == df2.sn_cen)),
              df1.dob_ccs == df2.dob_cen,
```

```
            df1.sex_ccs == df2.sex_cen,
            df1.pc_ccs == df2.pc_cen]

    # MK4 Alphaname, DOB, Sex, PC
    cond04 = [df1.alphaname_ccs == df2.alphaname_cen,
            df1.dob_ccs == df2.dob_cen,
            df1.sex_ccs == df2.sex_cen,
            df1.pc_ccs == df2.pc_cen]

    # MK5 FN Nickname, SN, DOB, Sex, PC
    cond05 = [df1.fn1_nickname_ccs == df2.fn1_nickname_cen,
            ((df1.sn1_ccs == df2.sn1_cen) | (df1.sn_ccs == df2.sn_cen)),
            df1.dob_ccs == df2.dob_cen,
            df1.sex_ccs == df2.sex_cen,
            df1.pc_ccs == df2.pc_cen]

    # MK6 FN Nickname, SN, AGE, Sex, UPRN
    cond06 = [df1.fn1_nickname_ccs == df2.fn1_nickname_cen,
            ((df1.sn1_ccs == df2.sn1_cen) | (df1.sn_ccs == df2.sn_cen)),
            df1.age_ccs == df2.age_cen,
            df1.sex_ccs == df2.sex_cen,
            df1.uprn_ccs == df2.uprn_cen]

    # MK7 Contained Name, DOB, Sex, PC+HN / UPRN
    cond07 = [df1.dob_ccs == df2.dob_cen,
            df1.sex_ccs == df2.sex_cen,
            ((df1.uprn_ccs == df2.uprn_cen) |
             ((df1.pc_ccs == df2.pc_cen) & (df1.house_no_ccs == df2.house_no_cen
))))]
            # Plus Contained Name]

    # MK8 Lev Fullname, DOB, Sex, PC
    cond08 = [levenshtein(df1.fullname_ns_ccs, df2.fullname_ns_cen) < 3,
            df1.dob_ccs == df2.dob_cen,
            df1.sex_ccs == df2.sex_cen,
            df1.pc_ccs == df2.pc_cen]
```

```
# MK9 Soundex FN, Soundex SN, DOB, Sex, PC
cond09 = [soundex(df1.fn1_ccs) == soundex(df2.fn1_cen),
          soundex(df1.sn1_ccs) == soundex(df2.sn1_cen),
          df1.dob_ccs == df2.dob_cen,
          df1.sex_ccs == df2.sex_cen,
          df1.pc_ccs == df2.pc_cen]

# MK10 Allowing for error only in Sex
cond10 = [((df1.fn1_ccs == df2.fn1_cen) | (df1.fn_ccs == df2.fn_cen)),
          ((df1.sn1_ccs == df2.sn1_cen) | (df1.sn_ccs == df2.sn_cen)),
          df1.dob_ccs == df2.dob_cen,
          df1.pc_ccs == df2.pc_cen]

# MK11 Allowing for error in name (strict Jaro)
cond11 = [# JARO(df1.fn1_ccs, df2.fn1_cen) > 0.9,
          # JARO(df1.sn1_ccs, df2.sn1_cen) > 0.8,
          df1.dob_ccs == df2.dob_cen,
          df1.sex_ccs == df2.sex_cen,
          df1.pc_ccs == df2.pc_cen]

# MK12 Allowing for error in Name (Levenstein)
cond12 = [lev_score('fn1_ccs', 'fn1_cen') > 0.60,
          lev_score('sn1_ccs', 'sn1_cen') > 0.60,
          df1.dob_ccs == df2.dob_cen,
          df1.sex_ccs == df2.sex_cen,
          df1.pc_ccs == df2.pc_cen]

# MK13 Lev FN-SN & SN-FN, DOB, Sex, PC
cond13 = [lev_score('fn1_ccs', 'sn1_cen') > 0.60,
          lev_score('sn1_ccs', 'fn1_cen') > 0.60,
          df1.dob_ccs == df2.dob_cen,
          df1.sex_ccs == df2.sex_cen,
          df1.pc_ccs == df2.pc_cen]

# MK14 FN2, SN, DOB, Sex, PC
cond14 = [df1.fn2_ccs == df2.fn2_cen,
```

```
            ((df1.sn1_ccs == df2.sn1_cen) | (df1.sn_ccs == df2.sn_cen)),
            df1.dob_ccs == df2.dob_cen,
            df1.sex_ccs == df2.sex_cen,
            df1.pc_ccs == df2.pc_cen]

    # MK15 Lev Edit Distance FN1, lev Edit Distance FN2, DOB, Sex, PC
    cond15 = [lev_score('fn1_ccs', 'fn1_cen') > 0.60,
            lev_score('fn2_ccs', 'fn2_cen') > 0.60,
            df1.dob_ccs == df2.dob_cen,
            df1.sex_ccs == df2.sex_cen,
            df1.pc_ccs == df2.pc_cen]

    # MK16 Swapped FNs, SN, DOB, PC
    cond16 = [((df1.fn1_ccs == df2.fn2_cen) | (df1.fn2_ccs == df2.fn1_cen)),
            ((df1.sn1_ccs == df2.sn1_cen) | (df1.sn_ccs == df2.sn_cen)),
            df1.dob_ccs == df2.dob_cen,
            df1.sex_ccs == df2.sex_cen,
            ((df1.uprn_ccs == df2.uprn_cen) |
             ((df1.pc_ccs == df2.pc_cen) & (df1.house_no_ccs == df2.house_no_cen
)))]

    # MK17 FN, DOB, Sex, (UPRN or PC + HN)
    cond17 = [((df1.fn1_ccs == df2.fn1_cen) | (df1.fn_ccs == df2.fn_cen)),
            df1.dob_ccs == df2.dob_cen,
            df1.sex_ccs == df2.sex_cen,
            ((df1.uprn_ccs == df2.uprn_cen) |
             ((df1.pc_ccs == df2.pc_cen) & (df1.house_no_ccs == df2.house_no_cen
)))]

    # MK18 FN Lev > 0.5, SN, DOB, Sex, (UPRN or PC + HN)
    cond18 = [lev_score('fn1_ccs', 'fn1_cen') > 0.60,
            ((df1.sn1_ccs == df2.sn1_cen) | (df1.sn_ccs == df2.sn_cen)),
            df1.dob_ccs == df2.dob_cen,
            df1.sex_ccs == df2.sex_cen,
            ((df1.uprn_ccs == df2.uprn_cen) |
             ((df1.pc_ccs == df2.pc_cen) & (df1.house_no_ccs == df2.house_no_cen
)))]
```

```
    # MK19 FN Bigram, SN, DOB, Sex, (UPRN or PC + HN)
    cond19 = [df1.fn1_ccs.substr(1, 2) == df2.fn1_cen.substr(1, 2),
              ((df1.sn1_ccs == df2.sn1_cen) | (df1.sn_ccs == df2.sn_cen)),
              df1.dob_ccs == df2.dob_cen,
              df1.sex_ccs == df2.sex_cen,
              ((df1.uprn_ccs == df2.uprn_cen) |
               ((df1.pc_ccs == df2.pc_cen) & (df1.house_no_ccs == df2.house_no_cen
))))]

    # MK20 FN, SN, PC, SEX, AGE
    cond20 = [((df1.fn1_ccs == df2.fn1_cen) | (df1.fn_ccs == df2.fn_cen)),
              ((df1.sn1_ccs == df2.sn1_cen) | (df1.sn_ccs == df2.sn_cen)),
              df1.age_ccs == df2.age_cen,
              df1.sex_ccs == df2.sex_cen,
              df1.pc_ccs == df2.pc_cen]

    # MK21 FN, SN, PC, Sex, Day, Month, Year within 10
    cond21 = [((df1.fn1_ccs == df2.fn1_cen) | (df1.fn_ccs == df2.fn_cen)),
              ((df1.sn1_ccs == df2.sn1_cen) | (df1.sn_ccs == df2.sn_cen)),
              df1.sex_ccs == df2.sex_cen,
              df1.day_ccs == df2.day_cen,
              df1.mon_ccs == df2.mon_cen,
              df1.pc_ccs == df2.pc_cen,
              (abs_(df1.year_ccs - df2.year_cen) < 11)]

    # MK22 FN, SN, PC, Sex, Year
    cond22 = [((df1.fn1_ccs == df2.fn1_cen) | (df1.fn_ccs == df2.fn_cen)),
              ((df1.sn1_ccs == df2.sn1_cen) | (df1.sn_ccs == df2.sn_cen)),
              df1.sex_ccs == df2.sex_cen,
              df1.year_ccs == df2.year_cen,
              df1.pc_ccs == df2.pc_cen]

    # MK23 Jaro FN, Jaro SN, Lev DOB < 2, Sex, UPRN
    cond23 = [# JARO(df1.fn1_ccs, df2.fn1_cen) > 0.8,
              # JARO(df1.sn1_ccs, df2.sn1_cen) > 0.8,
              levenshtein(df1.dob_ccs, df2.dob_cen) < 2,
```

```
            df1.sex_ccs == df2.sex_cen,
            df1.uprn_ccs == df2.uprn_cen]

    # MK24 Lev FN, Lev SN, Lev DOB < 2, Sex, UPRN
    cond24 = [lev_score('fn1_ccs', 'fn1_cen') > 0.60,
            lev_score('sn1_ccs', 'sn1_cen') > 0.60,
            levenshtein(df1.dob_ccs, df2.dob_cen) < 2,
            df1.sex_ccs == df2.sex_cen,
            df1.uprn_ccs == df2.uprn_cen]

    # MK25 FN Lev, SN Lev, Age diff < 2 years, Sex, UPRN
    cond25 = [lev_score('fn1_ccs', 'fn1_cen') > 0.60,
            lev_score('sn1_ccs', 'sn1_cen') > 0.60,
            abs_(df1.age_ccs - df2.age_cen) < 2,
            df1.sex_ccs == df2.sex_cen,
            df1.uprn_ccs == df2.uprn_cen]

    # MK26 FN Lev, SN Lev, Age diff < 2 years, Sex, PC + HN
    cond26 = [lev_score('fn1_ccs', 'fn1_cen') > 0.60,
            lev_score('sn1_ccs', 'sn1_cen') > 0.60,
            abs_(df1.age_ccs - df2.age_cen) < 2,
            df1.sex_ccs == df2.sex_cen,
            ((df1.pc_ccs == df2.pc_cen) & (df1.house_no_ccs == df2.house_no_cen)
)]

    # MK27 Lev FN, Lev SN, Sex Agree/Missing, DOB, UPRN
    cond27 = [lev_score('fn1_ccs', 'fn1_cen') > 0.60,
            lev_score('sn1_ccs', 'sn1_cen') > 0.60,
            ((df1.sex_ccs == df2.sex_cen) |
             (df1.sex_ccs.isNull()) | (df2.sex_cen.isNull())),
            df1.dob_ccs == df2.dob_cen,
            df1.uprn_ccs == df2.uprn_cen]

    # MK28 Multiple Moves (PC+HN / UPRN) (used to be 30)
    cond28 = [((df1.fn1_ccs == df2.fn1_cen) | (df1.fn_ccs == df2.fn_cen)),
            ((df1.sn1_ccs == df2.sn1_cen) | (df1.sn_ccs == df2.sn_cen)),
            df1.dob_ccs == df2.dob_cen,
```

```
            df1.sex_ccs == df2.sex_cen]
            # Plus Multiple moves]

    # MK29 FN1, SN1, DOB, Sex, Postcode (minus last character)
    cond29 = [df1.fn1_ccs == df2.fn1_cen,
            df1.sn1_ccs == df2.sn1_cen,
            df1.dob_ccs == df2.dob_cen,
            df1.sex_ccs == df2.sex_cen,
            df1.pc_substr_ccs == df2.pc_substr_cen]

    # MK30 FN1, SN1, DOB, Sex, Sector, House Number
    cond30 = [df1.fn1_ccs == df2.fn1_cen,
            df1.sn1_ccs == df2.sn1_cen,
            df1.dob_ccs == df2.dob_cen,
            df1.sex_ccs == df2.sex_cen,
            df1.pc_sect_ccs == df2.pc_sect_cen,
            df1.house_no_ccs == df2.house_no_cen]

    # MK31 FN, SN, DOB, PCA, Sex (CLERICAL MATCHKEY)
    cond31 = [((df1.fn1_ccs == df2.fn1_cen) | (df1.fn_ccs == df2.fn_cen)),
            ((df1.sn1_ccs == df2.sn1_cen) | (df1.sn_ccs == df2.sn_cen)),
            df1.dob_ccs == df2.dob_cen,
            df1.pc_area_ccs == df2.pc_area_cen,
            df1.sex_ccs == df2.sex_cen]

    # MK32 FN, SN, Sex, Age within 6, PC (CLERICAL MATCHKEY)
    cond32 = [lev_score('fn1_ccs', 'fn1_cen') > 0.80,
            lev_score('sn1_ccs', 'sn1_cen') > 0.80,
            abs_(df1.age_ccs - df2.age_cen) < 6,
            df1.pc_ccs == df2.pc_cen,
            df1.sex_ccs == df2.sex_cen]
```

Appendix B

Matchkeys (household matching)

Note that these matchkeys are correct as of 28/05/2021 but are still undergoing development and are subject to change when we have access to 2021 Census and CCS data.

```
# ------------------------------------------------------------------------
----- #

# ------ MK1: UPRN + One Common Surname + One Common Forename + One Common DOB ---
----- #

# ------------------------------------------------------------------------
----- #



# MK1

matches_1 = UPRN_Join.filter((HF.common_udf('fn_set_cen', 'fn_set_ccs') >= 1) &

                             (HF.common_udf('sn_set_cen', 'sn_set_ccs') >= 1) &

                             (HF.common_udf('dob_set_cen', 'dob_set_ccs') >= 1))



# ------------------------------------------------------------------------
------------ #

# ---- MK2: UPRN + Tenure + TYPE + RESCOUNT + (One Common Surname OR One Common Fo
rename) ----- #

# ------------------------------------------------------------------------
------------ #



# MK2

matches_2 = UPRN_TTR_Join.filter((HF.common_udf('fn_set_cen', 'fn_set_ccs') >= 1)
|

                                 (HF.common_udf('sn_set_cen', 'sn_set_ccs') >= 1))



# ------------------------------------------------------------------------
--------------------------- #

# ------ MK3: UPRN + 2 of Tenure, Type, Rescount + (One Common Surname OR Common F
orename) + Common DOB ------ #

# ------------------------------------------------------------------------
--------------------------- #



# MK3

matches_3 = CEN.join(CCS, on = [CEN.uprn_cen == CCS.uprn_ccs,

                             (((CEN.tenure_cen == CCS.tenure_ccs) & (CEN.typaccom_
cen == CCS.typaccom_ccs)) |
```

```
                                ((CEN.tenure_cen == CCS.tenure_ccs) & (CEN.no_resi_c
en == CCS.no_resi_ccs))   |
                                ((CEN.no_resi_cen == CCS.no_resi_ccs) & (CEN.typacco
m_cen == CCS.typaccom_ccs)))], how = 'inner')


matches_3 = matches_3.filter(((HF.common_udf('fn_set_cen', 'fn_set_ccs') >= 1) | (
HF.common_udf('sn_set_cen', 'sn_set_ccs') >= 1)) &
                                (HF.common_udf('dob_set_cen', 'dob_set_ccs') >= 1))



# ------------------------------------------------------------------------------
----- #
# ----------------- MK4: UPRN + One Common Surname + One Common DOB -------------
----- #
# ------------------------------------------------------------------------------
----- #



# MK4
matches_4 = UPRN_Join.filter((HF.common_udf('sn_set_cen', 'sn_set_ccs')   >= 1) &
                             (HF.common_udf('dob_set_cen', 'dob_set_ccs') >= 1))



# ------------------------------------------------------------------------------
--- #
# -------- MK5: UPRN + Max FN LEV > 0.80 + Max SN LEV > 0.80 + Common DOB --------
--- #
# ------------------------------------------------------------------------------
--- #



# MK5
matches_5 = UPRN_Join.filter((HF.max_lev_udf('fn_set_cen', 'fn_set_ccs') > 0.80) &
                             (HF.max_lev_udf('sn_set_cen', 'sn_set_ccs') > 0.80) &
                             (HF.common_udf('dob_set_cen', 'dob_set_ccs') >= 1))



# ------------------------------------------------------------------------ #
# ------ MK6: UPRN + Max FN JAR > 0.80 + Max SN JAR > 0.80 + Common DOB ------- #
# ------------------------------------------------------------------------ #
```

```
# MK6
matches_6 = UPRN_Join.filter((HF.max_jaro_udf('fn_set_cen', 'fn_set_ccs') > 0.80)
&
                             (HF.max_jaro_udf('sn_set_cen', 'sn_set_ccs') > 0.80)
&
                             (HF.common_udf('dob_set_cen', 'dob_set_ccs') >= 1))



# ------------------------------------------------------------------------------
------------------------------------------------ #
# -- MK7: UPRN + Max FN LEV > 0.60 + Max SN LEV > 0.60 + Common DOB (or 2 common D
OB if 4 + people) + TYPE + TENURE + RESCOUNT ----- #
# ------------------------------------------------------------------------------
------------------------------------------------ #



# MK7
matches_7 = UPRN_TTR_Join.filter((HF.max_lev_udf('fn_set_cen', 'fn_set_ccs') > 0.6
0) &
                                 (HF.max_lev_udf('sn_set_cen', 'sn_set_ccs') > 0.6
0) &
                                 (HF.common_udf('dob_set_cen', 'dob_set_ccs') >= 1
))


# 2 Common DOB required if 4+ people
matches_7 = matches_7.filter(~((HF.common_udf('dob_set_cen', 'dob_set_ccs') == 1)
& (matches_7.hh_size_cen >= 4) & (matches_7.hh_size_ccs >= 4)))



# ------------------------------------------------------------------------------
------------------------------------------------ #
# -- MK8: UPRN + Max FN JARO > 0.60 + Max SN JARO > 0.60 + Common DOB (or 2 common
DOB if 4 + people) + TYPE + TENURE + RESCOUNT --- #
# ------------------------------------------------------------------------------
------------------------------------------------ #



# MK8
```

```
matches_8 = UPRN_TTR_Join.filter((HF.max_jaro_udf('fn_set_cen', 'fn_set_ccs') > 0.
60) &
                                 (HF.max_jaro_udf('sn_set_cen', 'sn_set_ccs') > 0.
60) &
                                 (HF.common_udf('dob_set_cen', 'dob_set_ccs') >= 1
))


# 2 Common DOB required if 4+ people
matches_8 = matches_8.filter(~((HF.common_udf('dob_set_cen', 'dob_set_ccs') == 1)
& (matches_8.hh_size_cen >= 4) & (matches_8.hh_size_ccs >= 4)))



# ------------------------------------------------------------------------------
---------------------- #
# --- MK9: UPRN + (Max FN LEV > 0.80 OR Max SN LEV > 0.80) + Common DOB + TYPE + T
ENURE + RESCOUNT ----- #
# ------------------------------------------------------------------------------
---------------------- #



# MK9
matches_9 = UPRN_TTR_Join.filter(((HF.max_lev_udf('fn_set_cen', 'fn_set_ccs') > 0.
80) | (HF.max_lev_udf('sn_set_cen', 'sn_set_ccs') > 0.80)) &
                                 (HF.common_udf('dob_set_cen', 'dob_set_ccs') >=
1))



# ------------------------------------------------------------------------------
---------------- #
# ------ MK10: Single Person HH, UPRN, TYPE, TENURE, Equal DOB, Max FN or SN Lev >
0.50  --------- #
# ------------------------------------------------------------------------------
---------------- #



# MK10
matches_10 = UPRN_Join.filter(((HF.max_lev_udf('fn_set_cen', 'fn_set_ccs') > 0.50)
| (HF.max_lev_udf('sn_set_cen', 'sn_set_ccs') > 0.50)) &
                              (((UPRN_Join.tenure_cen == UPRN_Join.tenure_ccs)   &
(UPRN_Join.typaccom_cen == UPRN_Join.typaccom_ccs)) |
                               ((UPRN_Join.tenure_cen == UPRN_Join.tenure_ccs)   &
(UPRN_Join.no_resi_cen  == UPRN_Join.no_resi_ccs)) |
```

```
                                ((UPRN_Join.no_resi_cen == UPRN_Join.no_resi_ccs) &
(UPRN_Join.typaccom_cen == UPRN_Join.typaccom_ccs))) &

                                 (UPRN_Join.dob_set_cen == UPRN_Join.dob_set_ccs))


# Single Person HHs

matches_10 = matches_10.filter((matches_10.hh_size_cen == 1) & (matches_10.hh_size
_ccs == 1))




# ------------------------------------------------------------------------------
--------------- #

# -------- MK11: Single Person HH, UPRN, TYPE, TENURE, Equal DOB, FN or SN Jaro >
0.80  --------- #

# ------------------------------------------------------------------------------
--------------- #



# MK11

matches_11 = UPRN_Join.filter(((HF.max_jaro_udf('fn_set_cen', 'fn_set_ccs') > 0.80
) | (HF.max_jaro_udf('sn_set_cen', 'sn_set_ccs') > 0.80)) &

                               (((UPRN_Join.tenure_cen == UPRN_Join.tenure_ccs)   &
(UPRN_Join.typaccom_cen == UPRN_Join.typaccom_ccs)) |

                                ((UPRN_Join.tenure_cen == UPRN_Join.tenure_ccs)   &
(UPRN_Join.no_resi_cen  == UPRN_Join.no_resi_ccs)) |

                                ((UPRN_Join.no_resi_cen == UPRN_Join.no_resi_ccs) &
(UPRN_Join.typaccom_cen == UPRN_Join.typaccom_ccs))) &

                                 (UPRN_Join.dob_set_cen == UPRN_Join.dob_set_ccs))


# Single Person HHs

matches_11 = matches_11.filter((matches_11.hh_size_cen == 1) & (matches_11.hh_size
_ccs == 1))




# ------------------------------------------------------------------------------
--- #

# ------- MK12: Single Person HH, UPRN, TYPE, TENURE, Max FN or SN Lev > 0.80 ----
--- #

# ------------------------------------------------------------------------------
--- #
```

```
# MK12
matches_12 = UPRN_Join.filter(((HF.max_lev_udf('fn_set_cen', 'fn_set_ccs') > 0.80)
| (HF.max_lev_udf('sn_set_cen', 'sn_set_ccs') > 0.80)) &
                              (((UPRN_Join.tenure_cen == UPRN_Join.tenure_ccs)   &
(UPRN_Join.typaccom_cen == UPRN_Join.typaccom_ccs)) |
                               ((UPRN_Join.tenure_cen == UPRN_Join.tenure_ccs)   &
(UPRN_Join.no_resi_cen  == UPRN_Join.no_resi_ccs)) |
                               ((UPRN_Join.no_resi_cen == UPRN_Join.no_resi_ccs) &
(UPRN_Join.typaccom_cen == UPRN_Join.typaccom_ccs))))


# Single Person HHs
matches_12 = matches_12.filter((matches_12.hh_size_cen == 1) & (matches_12.hh_size
_ccs == 1))



# ------------------------------------------------------------------------------
----- #
# ------ MK13: PC + One Common Surname + One Common Forename + One Common DOB ----
----- #
# ------------------------------------------------------------------------------
----- #



# MK13
matches_13 = PC_Join.filter((HF.common_udf('fn_set_cen', 'fn_set_ccs') >= 1) &
                            (HF.common_udf('sn_set_cen', 'sn_set_ccs') >= 1) &
                            (HF.common_udf('dob_set_cen', 'dob_set_ccs') >= 1))



# ------------------------------------------------------------------------ #
# ----- MK14: PC + Tenure + TYPE + RESCOUNT + One Common FN, SN and AGE ------ #
# ------------------------------------------------------------------------ #



# MK14
matches_14 = PC_TTR_Join.filter((HF.common_udf('fn_set_cen', 'fn_set_ccs') >= 1) &
                                (HF.common_udf('sn_set_cen', 'sn_set_ccs') >= 1) &
                                (HF.common_age_udf('age_set_cen', 'age_set_ccs') >
= 1))
```

```
# ------------------------------------------------------------------------------
------------------------------------------------------------ #

# -------- MK15: PC + 2 of Tenure, TYPE, RESCOUNT + (One Common Surname OR Common
Forename) + Common DOB (or 2 common DOB if 3+ people) ----- #

# ------------------------------------------------------------------------------
------------------------------------------------------------ #



# MK15

matches_15 = CEN.join(CCS, on = [CEN.pc_cen == CCS.pc_ccs,

                                 (((CEN.tenure_cen == CCS.tenure_ccs) & (CEN.typaccom
_cen == CCS.typaccom_ccs)) |

                                  ((CEN.tenure_cen == CCS.tenure_ccs) & (CEN.no_resi_
cen == CCS.no_resi_ccs))   |

                                  ((CEN.no_resi_cen == CCS.no_resi_ccs) & (CEN.typacc
om_cen == CCS.typaccom_ccs)))], how = 'inner')


matches_15 = matches_15.filter(((HF.common_udf('fn_set_cen', 'fn_set_ccs') >= 1) |
(HF.common_udf('sn_set_cen', 'sn_set_ccs') >= 1)) &

                                (HF.common_udf('dob_set_cen', 'dob_set_ccs') >= 1)
)


# 2 Common DOB required if 3+ people

matches_15 = matches_15.filter(~((HF.common_udf('dob_set_cen', 'dob_set_ccs') == 1
) & (matches_15.hh_size_cen >= 3) & (matches_15.hh_size_ccs >= 3)))



# ------------------------------------------------------------------------------
------------------------------------------------------ #

# --- MK16: PC + (Max FN LEV > 0.80 OR Max SN LEV > 0.80) + Common DOB (or 2 commo
n DOB if 3+ people) + TYPE + TENURE + RESCOUNT ------- #

# ------------------------------------------------------------------------------
------------------------------------------------------ #



# MK16

matches_16 = PC_TTR_Join.filter(((HF.max_lev_udf('fn_set_cen', 'fn_set_ccs') > 0.8
0) |

                                 (HF.max_lev_udf('sn_set_cen', 'sn_set_ccs') > 0.8
0)) &
```

```
                                    (HF.common_udf('dob_set_cen', 'dob_set_ccs') >= 1
))


# 2 Common DOB required if 4+ people

matches_16 = matches_16.filter(~((HF.common_udf('dob_set_cen', 'dob_set_ccs') == 1
) & (matches_16.hh_size_cen >= 3) & (matches_16.hh_size_ccs >= 3)))



# ------------------------------------------------------------------------------
------------------------------------------------------ #

# ---- MK17: PC + (Max FN JARO > 0.80 OR Max SN JARO > 0.80) + Common DOB (or 2 co
mmon DOB if 3+ people) + TYPE + TENURE + RESCOUNT ---- #

# ------------------------------------------------------------------------------
------------------------------------------------------ #



# MK17

matches_17 = PC_TTR_Join.filter(((HF.max_jaro_udf('fn_set_cen', 'fn_set_ccs') > 0.
80) |

                                    (HF.max_jaro_udf('sn_set_cen', 'sn_set_ccs') > 0.
80)) &

                                    (HF.common_udf('dob_set_cen', 'dob_set_ccs') >= 1
))


# 2 Common DOB required if 3+ people

matches_17 = matches_17.filter(~((HF.common_udf('dob_set_cen', 'dob_set_ccs') == 1
) & (matches_17.hh_size_cen >= 3) & (matches_17.hh_size_ccs >= 3)))



# ---------------------------------------------------------- #

# ---- MK18: PC + All Surnames and All Forenames Match ------ #

# ---------------------------------------------------------- #



# MK18

matches_18 = CEN.join(CCS, on = [(CEN.pc_cen == CCS.pc_ccs) & (CEN.sn_set_cen == C
CS.sn_set_ccs) & (CEN.fn_set_cen == CCS.fn_set_ccs)], how = 'inner')



# ------------------------------------------------------------------------------
------- #
```

```
# --- MK19: PC, House No, Common FN, Common SN, Common Age, 2 of TYPE,TENURE,RESCO
UNT --- #

# ------------------------------------------------------------------------------
------- #



# MK19

matches_19 = PC_Join.filter((PC_Join.house_no_cen == PC_Join.house_no_ccs) &

                            (((PC_Join.tenure_cen == PC_Join.tenure_ccs)   & (PC_Jo
in.typaccom_cen == PC_Join.typaccom_ccs)) |

                             ((PC_Join.tenure_cen == PC_Join.tenure_ccs)   & (PC_Jo
in.no_resi_cen  == PC_Join.no_resi_ccs)) |

                             ((PC_Join.no_resi_cen == PC_Join.no_resi_ccs) & (PC_Jo
in.typaccom_cen == PC_Join.typaccom_ccs))) &

                             ((HF.common_udf('fn_set_cen', 'fn_set_ccs') >= 1) & (H
F.common_udf('sn_set_cen', 'sn_set_ccs') >= 1) &

                              (HF.common_age_udf('age_set_cen', 'age_set_ccs') >= 1
)))



# ------------------------------------------------------------------------ #

# ---------- MK20: UPRN + (Two of Common FN / SN / DOB / Age) -------- #

# ------------------------------------------------------------------------ #



# MK20

matches_20 = UPRN_Join.filter(((HF.common_udf('fn_set_cen', 'fn_set_ccs') >= 1)
& (HF.common_udf('sn_set_cen',  'sn_set_ccs')  >= 1)) |

                              ((HF.common_udf('fn_set_cen', 'fn_set_ccs') >= 1)
& (HF.common_udf('dob_set_cen', 'dob_set_ccs') >= 1)) |

                              ((HF.common_udf('fn_set_cen', 'fn_set_ccs') >= 1)
& (HF.common_age_udf('age_set_cen', 'age_set_ccs') >= 1)) |

                              ((HF.common_udf('sn_set_cen', 'sn_set_ccs') >= 1)
& (HF.common_udf('dob_set_cen', 'dob_set_ccs') >= 1)) |

                              ((HF.common_udf('sn_set_cen', 'sn_set_ccs') >= 1)
& (HF.common_udf('age_set_cen', 'age_set_ccs') >= 1)))



# ------------------------------------------------------------------------ #

# ----------------- MK21: UPRN + JARO FN + JARO SN + SIZE ------------------- #

# ------------------------------------------------------------------------ #
```

```
# MK21
matches_21 = UPRN_Join.filter((HF.max_jaro_udf('fn_set_cen', 'fn_set_ccs') > 0.80)
&
                              (HF.max_jaro_udf('sn_set_cen', 'sn_set_ccs') > 0.80)
&
                              (UPRN_Join.hh_size_cen == UPRN_Join.hh_size_ccs))


# ------------------------------------------------------------------ #
# ----------------- MK22: UPRN + SIZE + ALL AGES ------------------- #
# ------------------------------------------------------------------ #

# MK22
matches_22 = UPRN_Join.filter((UPRN_Join.age_set_cen == UPRN_Join.age_set_ccs) &
                              (UPRN_Join.hh_size_cen == UPRN_Join.hh_size_ccs))


# ------------------------------------------------------------------------- #
# -------- MK23: PC + HN + FLAT + (Common FN OR SN OR DOB) + SIZE  -------- #
# ------------------------------------------------------------------------- #


# MK23
matches_23 = PC_Join.filter((PC_Join.flat_no_cen == PC_Join.flat_no_ccs) &
                            (PC_Join.house_no_cen == PC_Join.house_no_ccs) &
                            (PC_Join.hh_size_cen  == PC_Join.hh_size_ccs) &
                           ((HF.common_udf('fn_set_cen', 'fn_set_ccs') >= 1) |
                            (HF.common_udf('dob_set_cen', 'dob_set_ccs') >= 1) |
                            (HF.common_udf('sn_set_cen', 'sn_set_ccs') >= 1)))


# ------------------------------------------------------------------ #
# -------- MK24: PC + FLAT + (Common FN AND DOB) + SIZE  -------- #
# ------------------------------------------------------------------ #
```

```
# MK24
matches_24 = PC_Join.filter((PC_Join.flat_no_cen == PC_Join.flat_no_ccs) &
                            (PC_Join.hh_size_cen == PC_Join.hh_size_ccs) &
                            (HF.common_udf('fn_set_cen', 'fn_set_ccs') >= 1) &
                            (HF.common_udf('dob_set_cen', 'dob_set_ccs') >= 1))


# ------------------------------------------------------------- #
# -------- MK25: PC + FLAT + (Common SN AND DOB) + SIZE  -------- #
# ------------------------------------------------------------- #


# MK25
matches_25 = PC_Join.filter((PC_Join.flat_no_cen == PC_Join.flat_no_ccs) &
                            (PC_Join.hh_size_cen == PC_Join.hh_size_ccs) &
                            (HF.common_udf('sn_set_cen', 'sn_set_ccs') >= 1) &
                            (HF.common_udf('dob_set_cen', 'dob_set_ccs') >= 1))


# ------------------------------------------------------------------------ #
# ------ MK26: Single Person HH + UPRN + Common SN + Max FN LEV > 0.40  ----- #
# ------------------------------------------------------------------------ #


# MK26
matches_26 = UPRN_Join.filter((PC_Join.hh_size_cen  == 1) &
                              (PC_Join.hh_size_ccs  == 1) &
                              (HF.common_udf('sn_set_cen', 'sn_set_ccs') >= 1) &
                              (HF.max_lev_udf('fn_set_cen', 'fn_set_ccs') > 0.40))


# ------------------------------------------------------------------------ #
# ------ MK27: Single Person HH + PC + HN + Common SN + Max FN LEV > 0.50  ----- #
# ------------------------------------------------------------------------ #
```

```
# MK27
matches_27 = PC_Join.filter((PC_Join.house_no_cen == PC_Join.house_no_ccs) &
                            (PC_Join.hh_size_cen  == 1) &
                            (PC_Join.hh_size_ccs  == 1) &
                            (HF.common_udf('sn_set_cen', 'sn_set_ccs') >= 1) &
                            (HF.max_lev_udf('fn_set_cen', 'fn_set_ccs') > 0.50))


# ------------------------------------------------------------------------ #
# ------ MK28: UPRN + COMMON AGE + JARO FN + JARO SN + NO FLAT NUMBERS  ----- #
# ------------------------------------------------------------------------ #


# MK28
matches_28 = UPRN_Join.filter((PC_Join.flat_no_cen.isNull() == True) &
                              (PC_Join.flat_no_ccs.isNull() == True) &
                              (HF.common_age_udf('age_set_cen', 'age_set_ccs') >=
1) &
                              (HF.max_jaro_udf('fn_set_cen', 'fn_set_ccs') > 0.80)
&
                              (HF.max_jaro_udf('sn_set_cen', 'sn_set_ccs') > 0.80)
)


# ------------------------------------------------------------------------ #
# ------ MK29: PC + HN + COMMON AGE + JARO FN + JARO SN + NO FLAT NUMBERS  ----- #
# ------------------------------------------------------------------------ #


# MK29
matches_29 = PC_Join.filter((PC_Join.house_no_cen == PC_Join.house_no_ccs) &
                            (PC_Join.flat_no_cen.isNull() == True) &
                            (PC_Join.flat_no_ccs.isNull() == True) &
                            (HF.common_age_udf('age_set_cen', 'age_set_ccs') >= 1)
&
                            (HF.max_jaro_udf('fn_set_cen', 'fn_set_ccs') > 0.80) &
                            (HF.max_jaro_udf('sn_set_cen', 'sn_set_ccs') > 0.80))
```

Appendix C
Images of the Clerical Matching System

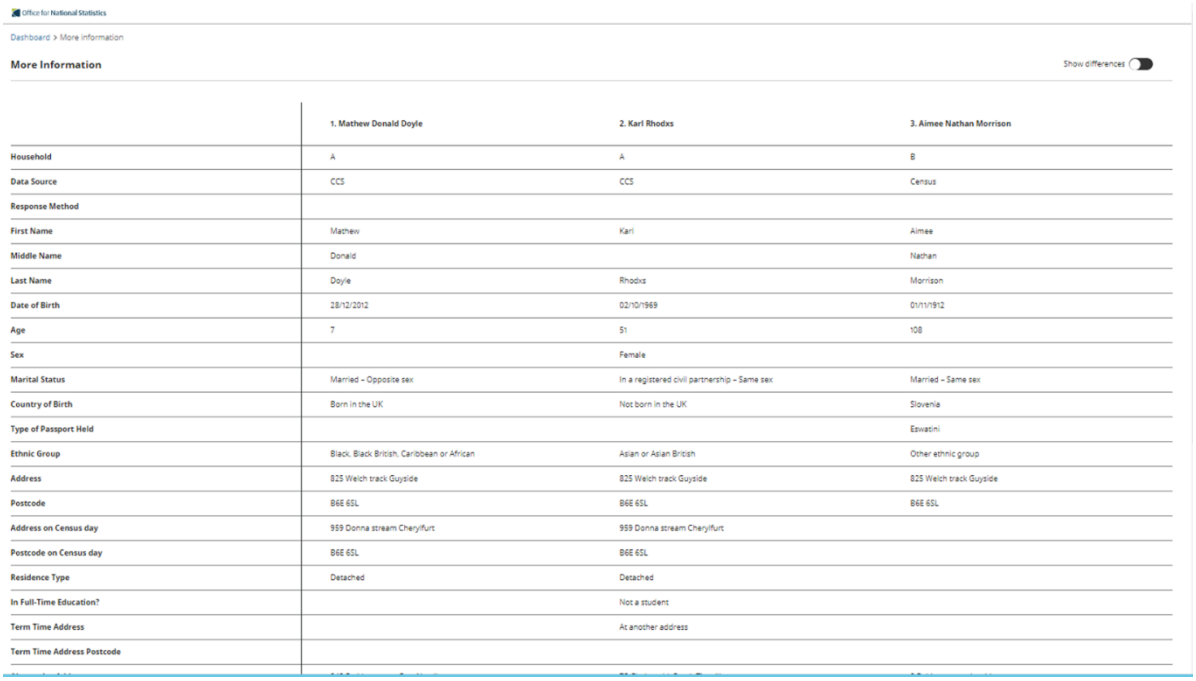
Note that these images are taken from the DevTest DAP environment and contain fake data records.

Individual matching screen

Here the clerical matcher can make person matches. Although a single pair of records are shown here, this screen can be used to display clusters of people if a non-unique match has been made.

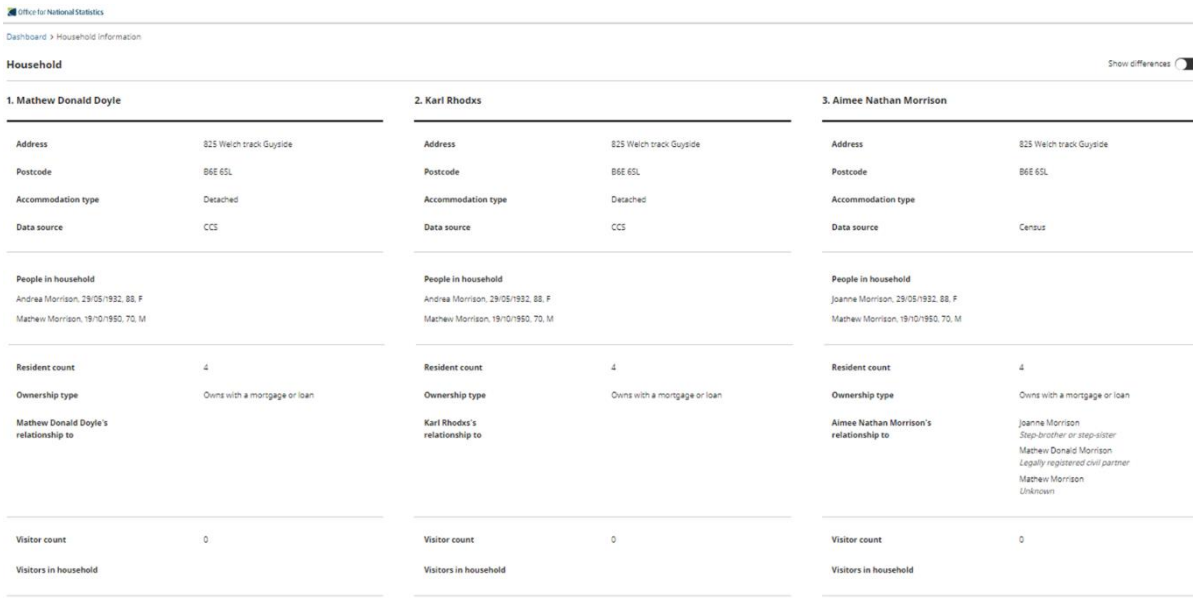
More Information screen – additional individual information

A clerical matcher can select between one and four individual records and see further information such as additional addresses, marital status, ethnicity etc



More information screen – household view

A clerical matcher can select between one and four individual records and see a household view that shows them who else was recording at the household. If there are other people in common between the households this provides evidence that the match is correct.



Office for National Statistics

Dashboard > Household information

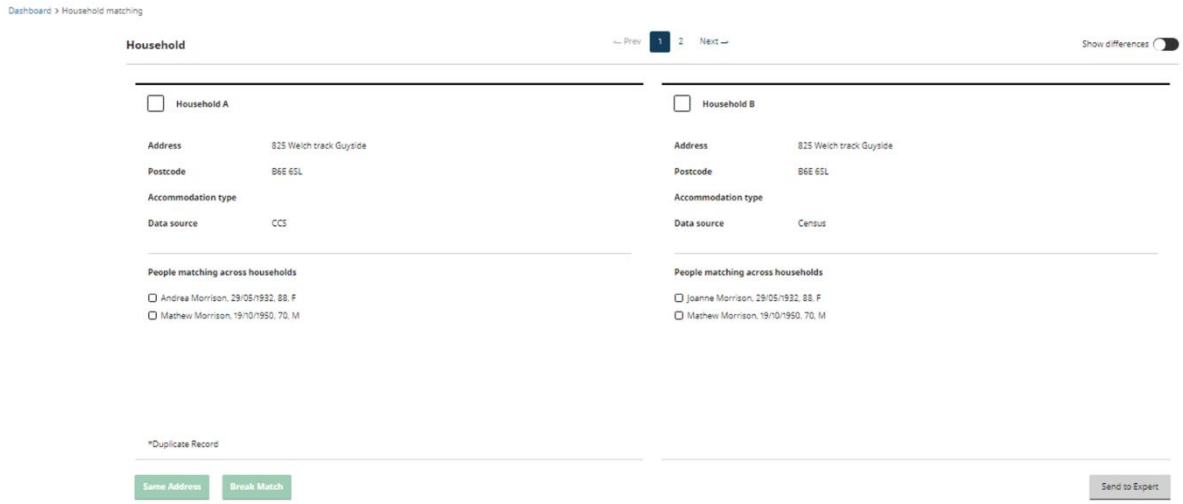
Household

Show differences

1. Mathew Donald Doyle		**2. Karl Rhodxs**		**3. Aimee Nathan Morrison**	
Address	825 Welch track Guyside	**Address**	825 Welch track Guyside	**Address**	825 Welch track Guyside
Postcode	B6E 6SL	**Postcode**	B6E 6SL	**Postcode**	B6E 6SL
Accommodation type	Detached	**Accommodation type**	Detached	**Accommodation type**	
Data source	CCS	**Data source**	CCS	**Data source**	Census
People in household	Andrea Morrison, 29/05/1932, 88, F Mathew Morrison, 19/10/1950, 70, M	**People in household**	Andrea Morrison, 29/05/1932, 88, F Mathew Morrison, 19/10/1950, 70, M	**People in household**	Joanne Morrison, 29/05/1932, 88, F Mathew Morrison, 19/10/1950, 70, M
Resident count	4	**Resident count**	4	**Resident count**	4
Ownership type	Owns with a mortgage or loan	**Ownership type**	Owns with a mortgage or loan	**Ownership type**	Owns with a mortgage or loan
Mathew Donald Doyle's relationship to		**Karl Rhodxs's relationship to**		**Aimee Nathan Morrison's relationship to**	Joanne Morrison *Step-brother or step-sister* Mathew Donald Morrison *Legally registered civil partner* Mathew Morrison *Unknown*
Visitor count	0	**Visitor count**	0	**Visitor count**	0
Visitors in household		**Visitors in household**		**Visitors in household**	

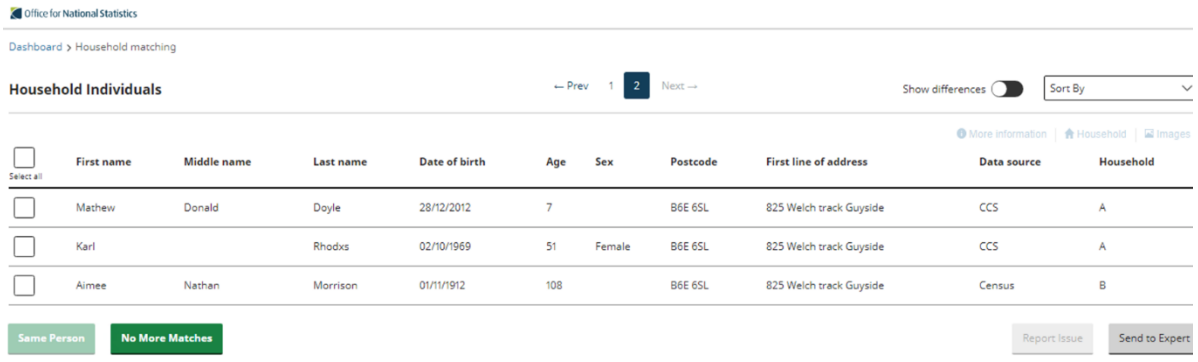
Household matching screen page 1

Here the clerical matcher can make household matches. They are given a view of the people in the household who are already matched. The clerical matcher can break a match if they think it is incorrect. On the second page (see below) the matcher is presented with all the unmatched people within these households and can make further person matches.



Household matching screen page 2

Here the clerical matcher can see if there are any additional person matches to be made between the two matched households A and B.



Appendix D
Matching outputs

1) CEN - CCS HOUSEHOLDS

	List of matched HH IDs (Census to CCS)	**Range**	
columns	Census_HH_ID	17 digit char string	Null for unmatched CCS records
	CCS_HH_ID	17 digit char string	Null for unmatched CEN records
	Match_Status	1 or 0	0 if no match, 1 if match
	Match_Score	integer	1 to N as appropriate if match made on matchkey 1-N, or 99 if match made by clerical. Null if Match_Status = 0
	Census_Postcode	string	Current postcode on Census. Null if Match_Status = 0
	CCS_Postcode	string	Current postcode on CCS. Null if Match_Status = 0
	Location_Flag	0,1, 2 or 3	0 if census postcode = ccs postcode, 1 if postcodes are contiguous, 2 both postcodes in CCS areas, 3 census postcode not in a CCS area. Null if Match_Status = 0
	Out_of_Scope_Flag	0,1,…,N	0 if in scope, 1,….,N otherwise (each non zero value represents a different reason for the individual CEN/CCS record being OOS) (see below for OOS reasons)
	Mover_Flag	1 or 0	1 if CCS HH has indicated that they moved house post census day. 0 otherwise.
	Duplicate_Flag_CEN	1 or 0	1 if 2 or more CEN HHs match to the same CCS HH, Null if Match_Status = 0. Otherwise 0
	Duplicate_Flag_CCS	1 or 0	1 if 2 or more CCS HHs match to the same CEN HH, Null if Match_Status = 0. Otherwise 0
	Split_Merge	1 or 0	1 if HH cluster is a split/merge candidate, 0 if it is not. Null if Match_Status = 0
	Cluster_Number	integer	For unique households this number will only appear once, for duplicates it will appear multiple times. Null if Match_Status = 0

2) CEN - CCS PEOPLE

(CMATCH-to-CENSAS--Cen2CCS_Per_Matchlist.csv)

	List of matched Person IDs (Census to CCS)	Range	
columns	Census_Resident_ID	20 digit char string	Null for unmatched CCS records
	CCS_Resident_ID	20 digit char string	Null for unmatched CEN records
	Match_Status	1 or 0	0 if no match, 1 if match
	Match_Score	integer	1 to N as appropriate if match made on matchkey 1-N, or 50 if made by probabilistic , or 99 if made by clerical. Null if Match_Status = 0
	Census_Postcode	string	Current postcode on Census. Null if Match_Status = 0
	CCS_Postcode	string	Current postcode on CCS. Null if Match_Status = 0
	Location_Flag	0,1, 2 or 3	0 if census postcode = ccs postcode, 1 if postcodes are contiguous, 2 both postcodes in CCS areas, 3 census postcode not in a CCS area. Null if Match_Status = 0
	Out_of_Scope_Flag	0,1,…,N	0 if in scope, 1,….,N otherwise (each non zero value represents a different reason for the individual CEN/CCS record being OOS)
	Mover_Flag	1 or 0	1 if CCS HH has indicated that they moved house post census day. 0 otherwise.
	Duplicate_Flag_CEN	1 or 0	1 if 2 or more CEN people match to the same CCS person, Null if Match_Status = 0. Otherwise 0
	Duplicate_Flag_CCS	1 or 0	1 if 2 or more CCS people match to the same CEN person, Null if Match_Status = 0. Otherwise 0
	Cluster_Number	integer	For unique people this number will only appear once, for duplicates it will appear multiple times. Null if Match_Status = 0

Individuals: Out of Scope (1 to 6 are flagged by clerics using the CMS), 7 is flagged automatically before matchin

3 is flagged automatically after matching (to allow for error in DOB to go through the CMS if necessary)

1 A match cannot be recorded because the record matches to a record where the individual is listed in the household but no individual information has been given
2 The record is missing all names and DOB
3 The individual is born after census day
4 A match cannot be recorded because the record matches to a visitor
5 Nonsense record (e.g. pets)
6 Other (for some other reason the clerical matcher can't match this record)
7 Student at a non-term time address

Households: Out of Scope

1 Empty household

Annex D: Shipsey, R. & Spakulova, I. (2021). Bayesian Approach to Sampling Applied to False Negative Assessment of Census to CCS Matching.

Introduction

In order to report on the accuracy of the census to CCS matching, we need to estimate the number of false positives (incorrect matches made) and false negatives (missed matches). False positives can occur at any stage of the matching process, both automatic and clerical. Estimation of the false positive rate is not covered in this paper. Here, we consider how to estimate the false negative rate.

A false negative (FN) in the census to CCS matching means that a CCS record did have a matching record in the census data, but we failed to make this match. Although a false negative can occur at any stage, the only stage at which they contribute to the overall false negative rate is at the pre-search stage. (See **Annex C** for a description of the different matching stages). This is because this is the final stage – in all previous stages, if a CCS record was not matched it would go onto the next stage and a match could still be found. However, at pre-search, if the record is not matched then this is the end of its journey – it is declared unmatchable. Therefore, in order to determine the false negative rate for the census to CCS matching we need to estimate how many false negatives are made at the pre-search stage. We will take a sample of CCS records not matched during pre-search and use clerical search to see if a match can be found in the census data. If no match is found, then the record is a true negative (TN). However, if a match is found then the record is a FN.

This paper discusses the size of the sample required in order to be tolerably sure that we have achieved our accuracy target regarding the number of FNs in our outputs.

Accuracy Targets

The census to CCS matching feeds into the dual system estimation (DSE) process for calculating the population estimate. See [Benton, 2015] for an introduction to DSE. Since DSE does not deal well with errors in matching, there are stringent accuracy targets in place.

- Less than 0.25% false negatives (missed matches) – this means that we find at least 99.75% of all the possible matches.
- Less than 0.1% false positives (incorrect matches) – this means that at least 99.9% of the matches that we make are correct

According to the 2011 Census to CCS gold standard, there were 649,944 true positive matches and 59,527 unmatched CCS records. Using these figures, assuming the accuracy in 2011 was as required, then

$$649{,}944 > 99.75\% \text{ of the matches}$$

i.e. if there were less than 1,629 false negatives then the matching would meet the accuracy requirement of a false negative rate of less than 0.25%.

Based on the 2011 Census and CCS data, we expect around 60,000 CCS records to be unmatched, and require less than 1,629 of these to be FNs. Thus, our target for FNs for the pre-search stage would be to achieve less than 2.73% FNs and this threshold will be used in the remainder of this paper.

The value of this threshold will vary depending on the number of matched and unmatched CCS records when matching is completed in 2021. We will denote this by θ_c which can be calculated as follows:

$$\theta_c = \frac{Allowed\ FN\ count}{Number\ of\ unmatched\ CCS\ records}$$

Where the *Allowed FN count* is the maximum number of missed matches such that the overall accuracy remains below the accuracy target of 0.25% false negatives. It can be calculated using

$$\begin{aligned} Allowed\ FN\ count &= 0.0025 \times Number\ of\ true\ links \\ &= 0.0025 \times (Number\ of\ matched\ CCS\ records + Allowed\ FN\ count) \\ &= 0.0025/0.9975 \times Number\ of\ matched\ CCS\ records \\ &= \frac{Number\ of\ matched\ CCS\ records}{399} \end{aligned}$$

Combining these, gives equation 1 which can be used in 2021 to calculate the value of θ_c.

Equation 1:

$$\theta_c = \frac{Number\ of\ matched\ CCS\ records}{399 \times Number\ of\ unmatched\ CCS\ records}$$

We need to take a sample of records unmatched after pre-search and from this estimate the probability of a FN at pre-search, $\hat{\theta}$, and a measure of how confident we are that the true probability of a FN at pre-search, θ, is less than θ_c. We will denote the size of this sample by N.

Bayesian Inference of a Binomial Proportion

We will use a Bayesian approach to sampling (see [Quantstat, 2021]) because this allows us to more easily interpret and directly estimate probabilities from the posterior distribution. It is therefore easier to quantify the uncertainty in our estimate of the false negative rate. We would expect the sample size for a Bayesian approach with an uninformed prior to be very similar to the frequentist approach to sample size calculations.

The problem of estimating the false negative rate in the census to CCS matching meets all the necessary assumptions for a Bayesian approach, namely:

- There are 2 outcomes – in our case a match is found (FN) or no match is found (TN)
- The trials are independent of each other – in our case, the fact that record A is a FN or a TN has no effect on whether record B is a FN or a TN. This is true because records A and B are CCS records and we search all of census for a match. So even if A matched to C and then B matched to C, the result for A has no impact on the result for B.
- The probability of a FN, θ, is constant over time (stationary). This is true since the number of FNs will be constant (we will not change matching decisions during the final QA process).

We will use a beta distribution, $beta(\alpha, \beta)$ to model our prior beliefs. Examples of the parameters (α, β) of the distribution are as follows and the distributions are shown in Figure 1.

- $beta(1,1)$ – we have no prior belief as to the value of θ (this prior distribution is sometimes referred to as an uninformative or flat prior distribution)
- $beta(3,100)$ – this is a cautious prior belief. The peak, $\hat{\theta}$, of the distribution occurs just below θ_c, but the distribution is quite wide, meaning that we cannot be confident that the true value of θ is less than θ_c

- $beta(30,1300)$ – this is a more optimistic prior belief. Again, the peak of the distribution occurs just below θ_c, but the distribution is narrower, meaning that we are more confident that the true value of θ is less than θ_c
- $beta(19,783)$ – these values of α and β could come from some observed data collected as pre-search is tuned. For example, if we took a sample of 800 records and found that 18 of them were FNs.

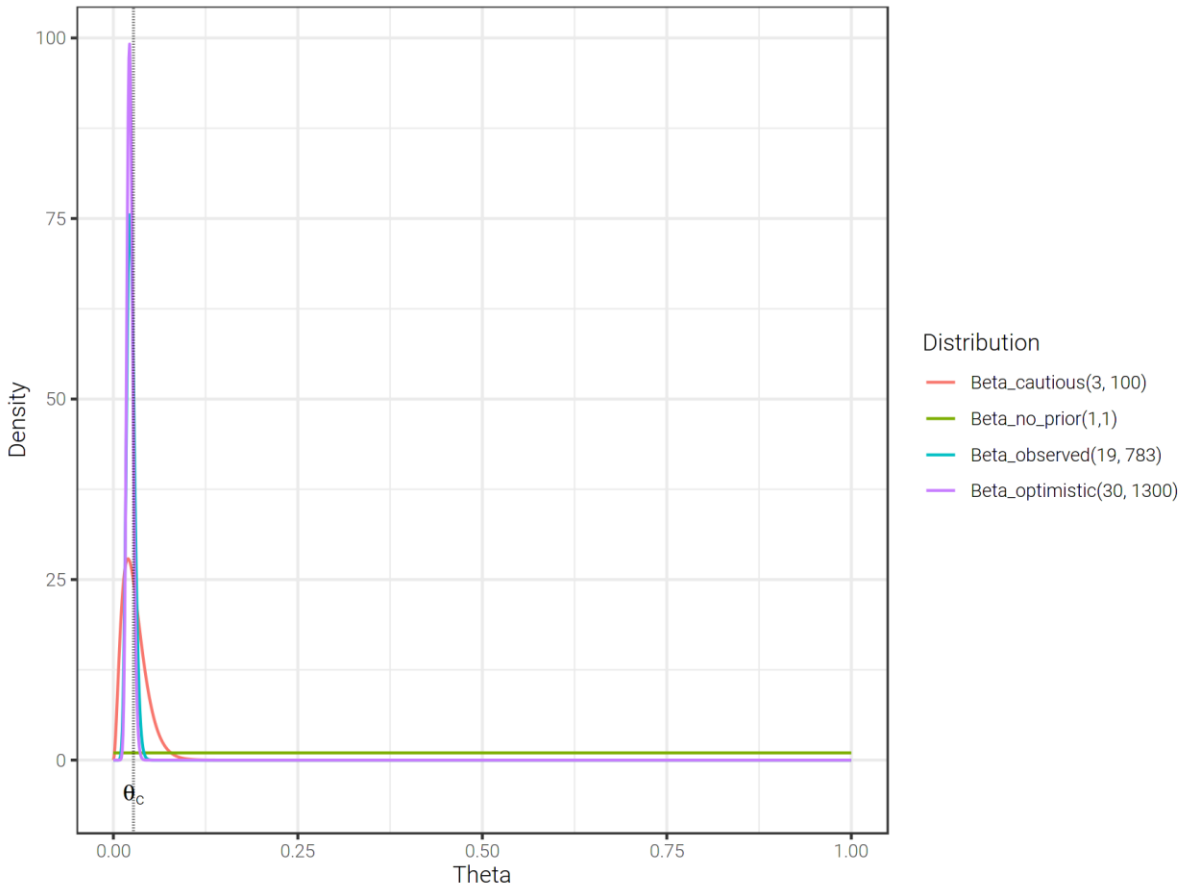


Figure 1. Beta distributions with varying parameters and FN probability threshold θ_c shown by the vertical dotted line

Once the matching of census to CCS is complete, we will pick a random sample of N CCS records that have not been matched and try to find a match for each of these using clerical search. The number of FNs in the sample (records for which a match is found) is denoted by z.

The data collected during the clerical search will allow us to update our estimate of $\hat{\theta}$ and our certainty that the true value of θ is less than θ_c.

The posterior belief is also modelled as a beta distribution with parameters α_1 and β_1 updated as follows:

Equation 2:

$$\alpha_1 = z + \alpha, \beta_1 = N - z + \beta$$

For example, suppose we consider the prior belief $beta(3,100)$ and then carried out clerical search on a sample of $N = 1{,}000$ records, finding that $z = 24$ (2.4%) of them were FNs. Our posterior belief would be modelled by $beta(27, 1076)$.

If we took a larger sample of N=2000 records and found that the same proportion of these were FNs z=48 (2.4%), then our posterior belief would be modelled by $beta(51, 2052)$. The graph in figure 2 shows that in both cases the new peak of the distribution is less than θ_c , however the width of the curve has narrowed more with the larger sample, meaning that we are more confident that the true value of θ is less than θ_c.

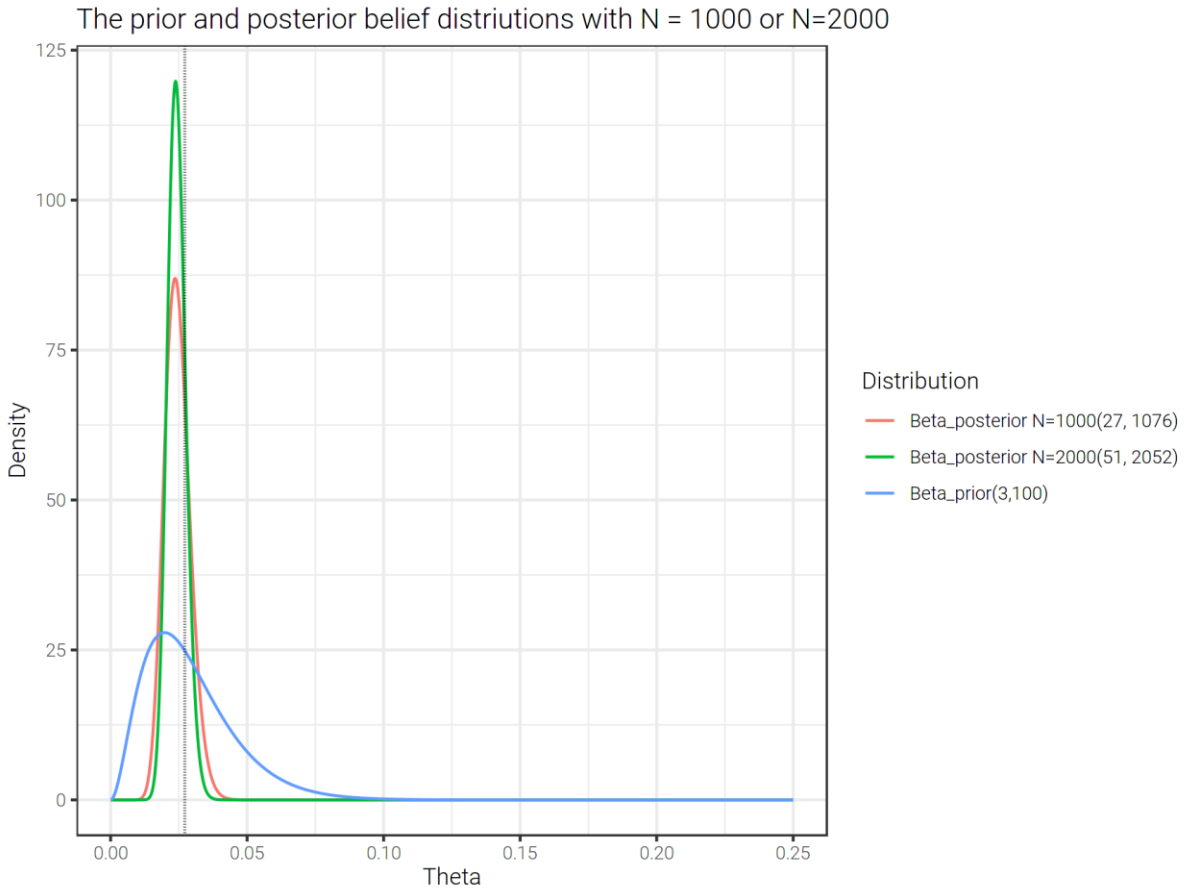


Figure 2. Cautious prior distribution and posterior distributions with sample sizes N=1000 and N=2000, in both cases assuming a pre-search FN rate of 2.4%.

How big does N have to be/How confident do we need to be?

From Equation 2 and Figure 2, we can see that both the size of the sample (N) and the number of FNs (z) have an effect on the posterior beta distribution. We need to choose N large enough to ensure that the width of the peak is narrow enough that most of the beta distribution is to the left of θ_c. However, we need to strike a balance between choosing large enough N but not requiring too much clerical searching as this is a time-consuming manual process.

We can choose an N such that, given our estimate of the FN rate $\hat{\theta}$, 95% of the posterior probability distribution is to the left of θ_c. Thus, with this sample size, assuming that our estimate of $\hat{\theta}$ is fairly accurate, we will be 95% sure that the true value of θ is less than θ_c.

Figure 3 shows how the sample size increases as the expected FN rate ($\hat{\theta}$) varies. If we expect $\hat{\theta}$ to be much less than θ_c then a small sample will confirm this with the required confidence. However, as $\hat{\theta}$ approaches θ_c, the sample size required increases rapidly until in effect we would have to find all of the FNs in order to be certain that θ is less than θ_c i.e. we would have to include all of the unmatched records in the sample.

The lines for the cautious prior belief and no prior belief are almost on top of each other, showing that having no prior belief is not much worse than having a cautious prior belief.

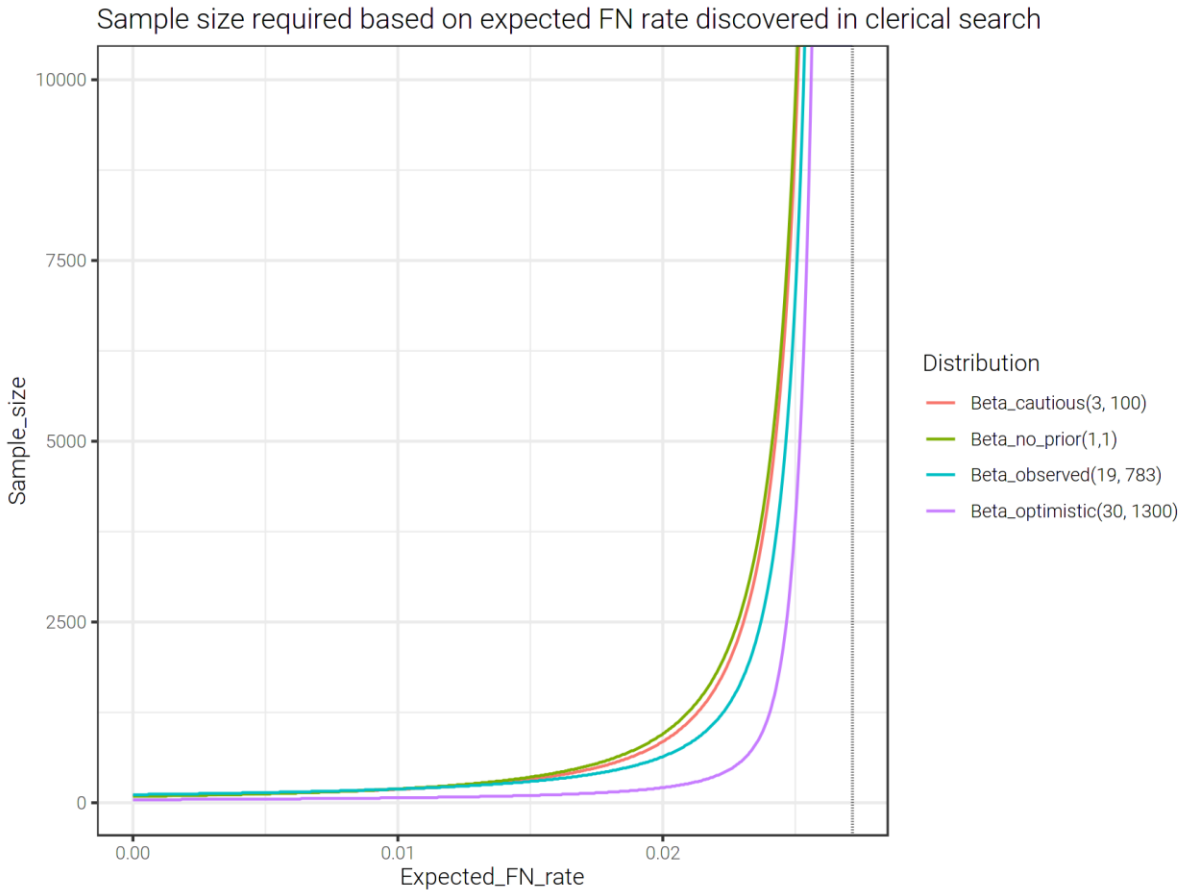


Figure 3. Sample size required to achieve 95% level of confidence given varying expected false negative rate.

Table 1. shows the sample size required to achieve a 95% level of confidence given varying numbers of FNs and depending on which prior distribution we use. Given that we will be able to form a prior belief using observed data as the pre-search algorithm is fine-tuned, it is perhaps the final column that gives the most realistic sample sizes. We should thus allow enough time to sample around 2,000 records. However, the sample size required may be smaller than this. Note that if our observed data informs us that the expected pre-search FN rate is very close to θ_c, then we will tune the parameters of the algorithm to improve this before continuing with pre-search. If it proves to be impossible to get pre-search working to within the required accuracy levels, then we will in any case revert to using clerical search for all of the unmatched CCS records, and this QA of pre-search will no longer be required.

Total number of FN	**True FN % rate θ assuming 60,000 unmatched records**	**Sample size no prior belief**	**Sample size cautious prior belief**	**Sample size optimistic prior belief**	**Sample size observed prior belief**
1000	1.67	470	420	120	360
1100	1.83	640	570	150	460
1200	2.00	950	850	210	640
1300	2.17	1580	1410	330	1000
1350	2.25	2170	1950	450	1370
1400	2.33	3170	2890	720	2020
1500	2.50	9730	9140	3890	7000
1600	2.67	Entire set	Entire set	Entire set	Entire set

Table 1. Examples of required samples sizes depending on our prior belief and estimate of the FN rate

Which prior belief to use

When we first start using the pre-search algorithm to make matches/declare records unmatchable, we will take a sample of 800 records that have been declared unmatchable and see if a match can be found using clerical search. Any records for which a match is found are false negatives (although at this stage their status can be corrected so that they become true positives). If the number of false negatives found is small enough, pre-search will continue with the algorithm unchanged. However, if the number of false negatives is not small enough, then the algorithm will be tuned, making use of the new information gained by analysis of the false negatives. Now all the unmatched CCS records will again go through the pre-search algorithm and another sample of 800 unmatchable records drawn. This process will continue until the number of false negatives is small enough. We therefore have to define what 'small enough' means in this context.

The exact number of false negatives that is 'small enough' will depend on the value of θ_c, which varies depending on the total number of unmatched and matched CCS records. Based on the 2011 data and value of θ_c, we can see from Table 1, that in order to keep the sample size for quality assurance to a realistic amount, we would ideally find 18 or less false negatives in our sample of 800. This is based on using a prior beta distribution of $\beta(19{,}783)$ which is the posterior distribution when starting with a

prior distribution of $\beta(1,1)$ i.e. the equivalent of having no prior belief, and then observing 18 FNs out of a sample of 800.

Note that if we found 19 FNs in the sample of 800 and changed our new prior belief according, then the final sample size would be around 3,200. If we found 20 FNs then the final sample size increases to around 9,000.

The initial sample size of 800 is chosen as a result of balancing the cost of clerical search with the risk of not obtaining a significant conclusion. If the pre-search performance on 2021 data is comparable to 2011 data, then the probability that we will have to repeat the sampling (i.e. there are more than 18 false negatives found in the sample of 800 unmatched CCS records) is less than 5%.

Conclusion

Provided that the pre-search algorithm works as expected, and the number of false negatives at the pre-search stage is less than the maximum acceptable number of false negatives, a sample of size around 2000 should be large enough to establish with a 95% confidence that the true value of the false negative rate is indeed less than maximum allowable false negative rate. However, if the pre-search algorithm appears to be working less well than expected, and we are very close to reaching the maximum allowable number of false negatives, then we will not be able to detect this using a sampling method and will instead have to use clerical search to classify all unmatched CCS records.

References

Benton, P. (2015) Trout, Catfish and Roach The beginner's guide to census population estimates available from https://fliphtml5.com/wihc/qghm/basic (accessed 5/1/2021)

Quatstat. Bayesian Inference of a Binomial Proportion – The Analytical Approach. Available from https://www.quantstart.com/articles/Bayesian-Inference-of-a-Binomial-Proportion-The-Analytical-Approach/#:~:text=Bayesian%20Inference%20of%20a%20Binomial%20Proportion%20-%20The,about%20uncertainty%20as%20new%20evidence%20came%20to%20light. (accessed 1/2/2021)

Annex E: Williams, K. (2020). All day matching Testing Report – July 2020.

All Day Matching Testing Report – July 2020

Contents

Executive summary

Cognitive and usability testing on the proposed clerical matching system for Census 2020 was conducted in July 2020 by Census and Population Statistics Hub in Methodology. Five clerical matchers (CM) were observed remotely completing a full day of clerical matching, as part of the census processing rehearsal, and then took part in a research interview. These findings incorporate the qualitative observations and interview and are supported by quantitative data [2]on the speed of matches. Additional data on accuracy can be incorporated when available. The test went very well despite the challenges of working remotely, with several key findings and recommendations in relation to optimal shift patterns, matching system, training and break length for CM for the 2021 Census.

Key findings

- CM would have liked more flexibility in the matching schedule. Typically, matchers would have preferred a longer lunch break with shorter morning and afternoon breaks, and it was found a break after one hour in the morning interrupted flow.
- Speed of matching varied by matching type and matcher; some increased in speed after lunch, whilst others decreased in speed.
- There were key usability issues which caused frustration for CM and decreased speed. These included inconsistencies in the highlighting feature, problems with names and 'household' data loading, losing positioning when navigating in and out of 'household' and 'more information' and the process of moving between two pages for 'household matching'.[3]
- Whilst training was found to be clear and thorough, CM did not feel confident to ask questions which resulted in inaccurate use of the system and lack of understanding in some instances, particularly in relation to 'household matching'. There were also inconsistencies in the use of 'send to expert' and 'report issue' buttons.

Key recommendations

[2] Given the small sample size, inference and significance cannot be drawn from the quantitative findings they are used to support the qualitative findings.

[3] Quantitative data in relation to accuracy of matches should be analysed to further inform these findings.

- Allow matchers flexibility in their daily schedule or implement a new advised break schedule: New advised break schedule: breaks every 90 minutes in the morning, 45-60 minute lunch break and breaks every 60 minutes in the afternoon.
- Resolve usability issues in relation to the highlighting feature, data loading, moving between two pages for 'household matching' and navigating in and out of 'household' and 'more information'.
- Clarify when to 'send to expert' vs. 'report issue' and check understanding in training.
- Replicate the office environment for matchers as much as possible during matching i.e. open skype call between matchers to discuss difficult matches and opportunities to ask questions.
- Consider practical sessions where clerical matching experts observe matchers prior to starting matching, particularly on 'household' matching, to resolve any misconceptions/ usability issues.

2.0 Research Design

- This paper presents findings and recommendations from Census and Population Statistics Hub Methodology's cognitive and usability testing on the Census clerical matching rehearsal. The testing was carried out in the form of remote all-day observations and cognitive interviews.
- Remote testing was conducted over skype, where 5 CM shared their screens and cameras for the full day, so the interviewers could observe their work.
- At the end of the day, interviewers conducted a cognitive interview with the CM.
- The CM had received training on clerical matching as part of the census matching rehearsal. It is important to note that due to unavoidable delays in the matching system/ data, training took place weeks before this research was conducted, which may have impacted upon findings.
- Matchers were randomly assigned to complete one of three types of matching during the day of being observed; individual matching, household matching or pre-search.
- Matchers were also randomly assigned to one of two conditions;
    - Condition A: matchers determined their own break times and lengths of breaks.
    - Condition B: matchers followed a set schedule of breaks (15-minute break every hour with a 45-minute lunch break replacing the second break).

3.0 Breaks

3.1 Qualitative Findings

Condition A (breaks determined by matcher, rather than imposed after an hour)

- It varied when CM took breaks;
    - Some took breaks consistently throughout the day i.e. after 1 hour.
    - Others took breaks inconsistently; they did not take a morning break but took an extended lunch and an afternoon break. It was noted that a morning break was not necessary for them and interrupted flow, however they were grateful for an afternoon break.
- Accuracy of work typically did not appear to change before/ after a break, however fatigue levels appeared to increase after lunch, which did appear to decrease speed of matching in some instances. This was irrespective of whether CM determined their own lunch or not.
- It was noted that the frequency and length of breaks taken was sufficient for them and there were no factors which stopped them in taking breaks, they simply decided when they felt they needed one i.e. for a drink/ lunch/ break away from screen.
- Observations showed that matchers looked alert for first half an hour but then displayed signs of boredom (however this initial alertness may have been observer effect).

Condition B (breaks as per schedule)

- CM did not naturally adhere to the advised schedule and had to be prompted to take the breaks. They typically stuck to the time of these prompted breaks, unless they had to take a phone call or answer the door (which happened 1-2 times during the day).
- It was noted that they would have preferred to decide their own breaks, as sometimes sticking to a schedule interrupted flow. For example, it was felt that in the morning, a break after an hour was ***"too soon"*** and they would have preferred to match for an hour and a half. However, in the

afternoon, matchers were grateful for the afternoon break after an hour of matching as they felt ***"tired"*** perhaps due to the repetitive nature of the task. Task switching could therefore be considered to provide variety to the day. However, we should first look at the effect of fatigue in the afternoon on accuracy once the data is available.

- Despite being given 15 minutes break, CM typically felt that 10 minutes break would have been long enough and would have preferred this with an extended lunch, especially when working from home.
- Observations showed that more breaks are needed in the afternoon as CM appeared tired. There also appeared to be a decrease in speed around this point.

Other findings

- Views on the length of day varied;
    - Some found the day too long and would have preferred to do more days with shorter periods of matching.
    - Others found the time spent matching appropriate.
- The only break which appeared to influence speed was lunch. It appeared that there were instances where CM speed decreased after lunch. Matchers also noted they became fatigued after this time, however suggested this may vary by individual and that this was typical of their normal workday where their productivity is better in the morning.
- Matchers who determined their own break lengths were slower at matching (average speed of 19.5 clusters per hour) compared to those who followed the break schedule (average speed of 36.1 clusters per hour). However, based on the small sample it is difficult to draw firm conclusions.

3.2 Quantitative Findings

- The average lunch break for CM (condition A and B) was 50 minutes, longer than the advised 45-minute lunch break. CM noted that they would have preferred longer. This suggests a longer lunch break should be factored into the clerical matching schedule.
- In condition A, where matchers decided their own breaks, they continued matching for an average of 31 minutes longer before taking their first break, when compared to condition B. They felt they did not need to take a break any sooner.
- In condition A, where matchers decided their own morning and afternoon breaks, they took an average of a 9-minute break, whereas the matchers who were on the break schedule took an average morning and afternoon break of 18 minutes, which they noted was too long. This suggests 10 minutes would be sufficient.
- Matches who were prompted to take breaks according to the advised schedule (condition B) spent longer taking breaks in a day (average total time of breaks 1 hour 24 minutes) than matchers who determined their own breaks (condition A- average total time of breaks 1 hour 11 minutes). This suggests that the advised breaks did not support optimal working for matchers as it decreased time spent on task, compared to matchers determining their own breaks.

3.3 Recommendations

- Qualitative and quantitative findings suggest matchers should be given some flexibility in determining their own breaks and break lengths.
- However, if implementing an advised break schedule;
    - Allow an hour and a half of matching in the morning before a break, opposed to an hour. This is because it typically took matchers between 10-30 minutes to get into a "flow", so felt being interrupted so soon after that for a break is not productive. Those in condition A who determined their own breaks also did not need a break after 1 hour in the morning.
    - Extend the lunch break to 1 hour and shorten the morning and afternoon breaks to 10 minutes.
    - Keep the break schedule within the afternoon, with a break after 1 hour to avoid fatigue.
    - Consider starting the matching day earlier as matchers appeared and felt more alert in the morning. However, it is important to note that this may depend on the matchers preferred

working schedules. A task switch could be trialled for CM who lose concentration in the afternoon.
    - New advised break schedule: breaks every 90 minutes in the morning, 45-60 minute lunch break and breaks every 60 minutes in the afternoon.

4.0 Speed

4.1 Qualitative Findings

- Matchers appeared to be slower within the first 10-30 minutes of the day, until they got into a ***"flow".*** In some instances, matchers noted that this was because they had forgotten the training.
- There were aspects of the clerical matching system which appeared to decrease the speed of matchers;
    - **Issues with the 'more information' and 'household' buttons**;
        - There were instances where the CM would have to select 'more information' to load and compare names, as the name did not always appear on the main matching screen.
        - It was noted that not all information associated with a cluster (a set of records which a decision is being made on) closed once a decision was made. This meant matchers had to spend time closing themselves.
        - Matchers frequently switched between the 'household' and 'more information' buttons. Each time, they returned to the main page, which decreased speed.
        - Matchers were observed manually dragging open the 'more information' and 'household' pages in order to view all records information.
    - **Issues with the highlighting feature;**
        - There was no option to show difference on 'break people match' of household section. Matchers would have liked this available.
        - Having highlighting on hid middle names in some instances on the 'pre search' task, meaning CM had to turn matching off in order to compare full names, decreasing speed.
        - Highlighting appeared for full addresses for CCS and Census records as CCS contained commas and Census did not.
        - The highlighting function did not stay visible when matchers scrolled down to view more records, therefore decreasing speed in making decisions.
    - When CM were working on the pre-search task, they frequently forgot which records they had already viewed and would open the same record multiple times in error. Matchers noted that this is because there was no option to eliminate records which they had already viewed.
    - After selecting 'same person' on page 2 of the household system, CM were automatically taken back to page 1. When matchers felt there were still records waiting to be matched, they would return to page 2 manually. This added burden to the CM and decreased their speed.
    - The 'sort by' feature did not stay visible when scrolling down to view all records.
- Speed of making a decision appeared to decrease in relation to the amount of information missing and number of records visible. However, the speed of CM appeared to increase when consecutive clusters were similar, for example, if they had a similar number of records to match or similar types of missing information/ potential errors. Matchers noted this was because they got into a good ***"flow".***
- Speed appeared to decrease when matchers used the 'send to expert' feature, as they were considering what to include in the write in box. Speed varied based on the amount of information matchers provided in this section, for example some matchers would simply note "need more info" whilst others would provide in depth explanations of why they considered it a potential match and what additional info they would have liked.

4.2 Quantitative Findings

- Speed of matching varied significantly between matchers and matching task (see Table 1). For example, on the day of being observed, CM05 had the fastest speed, at an average speed of 53.7 clusters per hour (individual matching) and CM02 had the slowest speed, at an average speed of 17.7 clusters per hour (pre search matching), excluding breaks.
- This speed appeared to be reflected within the full matching rehearsal, across days, where CM05 consistently worked at a faster pace, irrespective of matching type. CM05 also made less 'send to expert' decisions than other matchers. When they did 'send to expert' they did not provide much detail in the write in. It is important to check the accuracy of this matchers decisions before drawing any conclusions.
- Speed of matching typically decreased slightly immediately after lunch. However, it varied whether the average speed of clerical matchers remained at a slower pace throughout the afternoon. Some showed an increase in the afternoon, whilst others showed a decrease in the average speed in the afternoon (as shown in Table 1). This suggests that optimal times of working is dependent on the individuals preferred working patterns opposed to fatigue on the matching task.
- The average speed of matching per hour for pre search matching remained consistent, irrespective of matcher, at an average of 17.85 clusters per hour. This speed was maintained throughout the day and appeared to reflect the number of records on the page.
- The speed of clusters per hour for household matching varied based on matcher. On the days of observation, there was a difference of 9.3 matches per hour between matchers (CM01 completed 21.2 matcher per hour and CM05 completed 30.5 matches per hour). This reflects usability observations, which showed that CM01 did not interact with the household system accurately, whereas CM05 did, therefore increasing the speed of their matching.
- CMs typically took 10-20 minutes to get into a matching flow at the start of the day, indicated by a decreased speed between 10-10:20am before a steady increase in speed thereafter.

Table 1: Average speed of matching/ hour for each clerical matcher on day of observation (excluding break times)

	CM01	CM02	CM03	CM04	CM05
Type of matching	Household	Pre search	Household	Pre search	Individual
Condition	A	A	B	B	B
Average speed all day (decisions per hour)	21.2	17.7	36.5	18	53.7
Average speed before lunch (decisions per hour)	21.6	17.3	42.8	20.2	47.58
Average speed after lunch (decisions per hour)	20.7	18.3	30.5	15.8	58.42

Note: Red indicates a decrease in speed after lunch. Green indicates an increase in speed after lunch. Condition A = breaks determined by matcher, condition B = breaks according to advised schedule

4.3 Recommendations

- Consider giving a brief morning 'refresh' on the type of matching being completed that day.
- Aim to resolve usability issues within the system;
    - Keep the highlighting and sort by functions visible when scrolling down on a page.
    - Ensure names load without having to select 'more information'.
    - Where there are multiple records, consider 'greying out' records which have already been reviewed and eliminating those which are deemed as not a match.
    - Have an option to click through to 'more information' from the 'household' page, to avoid having to return to the main interface each time.
    - Within the household system, require CM to select 'no more matches' before returning them to page 1 of the interface. This would avoid a decrease in speed where CM have to return to page 2 each time they want to make a new match.
    - Have the 'household' and 'more information' pages open in full screen.

- If possible, put similar clusters (i.e. similar number of records with similar missing information) consecutively to improve productivity.
- Consider practical sessions where clerical matching experts observe matchers, particularly on 'household' matching to resolve any misconceptions/ usability issues as misunderstandings here appeared to have a significant impact on matching speed and likely accuracy.
- Clarify when and how much detail to provide when using the 'send to expert feature' as this appeared to impact speed.
- Factor in a 10-15-minute refresh phase in the matching schedule, at the beginning of each day and immediately after lunch, where clerical matchers will be slower as they get into "flow".
- Allow flexibility in working patterns, as some clerical matchers work at a better pace in the morning, whilst others work at a better pace in the afternoon.

5.0 Training/ resources

It is important to note that due to unavoidable delays in the data/ DAP system, training was delivered weeks prior to the census matching rehearsal, which may have impacted the findings.

5.1 Findings

- Overall, the training was found to be good, clear, thorough, helpful and sufficient. However, it was noted that there was a lot to take in within a short time frame and therefore felt slightly rushed.
- It was noted that they would have liked more hands-on opportunities to try matching themselves before starting the rehearsal, to remove the doubt that they were doing it inaccurately.
- In some instances, CM noted that they had forgotten their training. This caused inaccurate work for 'household matching'. In these instances, CM did not look at resources as they said they ***"didn't have time".***
- The examples, scenarios and videos given in training were found particularly useful; they liked being able to see actual examples of matching and learning how they would be resolved. However, it was noted that in some instances the training would require the matcher to ***"imagine"*** a situation, which was found to be confusing.
- Whilst matchers understood that training over skype was necessary, they would have preferred face to face training as they would have felt more able/ confident to ask questions. They would have also liked more opportunities to have their understanding checked within the training.
- CM noted that they would have liked the office environment replicated as much as possible, for example, having an open skype call with other matchers to ask them questions about difficult clusters, as you would with asking a colleague in the office.
- There was a desire for more training on which variables to pay particular attention to when matching, for example, have the trainer highlight the variables where error commonly occurs.
- CM did not always recall what 'pre search' matching was, suggesting training may not have been sufficient on this type of matching. CM did not always interact with the 'household' system accurately, suggesting training may not have been sufficient on this type of matching.
- Resources were found to be useful and sufficient, however it varied whether and how much matchers used the extra resources they were given;
    - Some did not look at the resources at all. They said they did not have time to do this. In these instances, it was suggested a shorter FAQ document could be given.
    - Some looked at certain resources which they felt were useful. For example, they used the name spreadsheet for foreign names when matching and used the flow chart to decide when to send to expert.
- It was noted that the name spreadsheet was not user friendly. There were also some issues with the search feature freezing, however this may have been specific to the laptop used.
- Despite CM stating a preference to have their queries to be answered straight away, opposed to sending to expert multiple times for the same query, matchers did not appear to ask questions to an expert directly via email or skype. Some noted that they should be more proactive themselves in

asking questions but did not feel confident to do so. They said they provided feedback in their daily feedback forms and would use the 'send to expert' button.

5.2 Recommendations

- Incorporate more practical aspects within the training, for example sessions where an expert observes a matcher for a set period to provide them with an opportunity to ask questions and correct misconceptions/ usability issues ahead of matching.
- If there are gaps between training and matching, provide refresh sessions on specific types of matching on the mornings before matching. This would help avoid matchers forgetting what is expected of them for each type of matching.
- Consider spending more time in training on how to interact with the 'household matching' system and what 'pre search' matching is.
- Where possible, use exact examples, opposed to getting the matchers to imagine situations. This should avoid matchers getting confused/ lost.
- Provide FAQ's to CM; these could be updated daily as new questions come in from clerical matchers, to save matchers sending to expert as frequently.
- Consider having matching expert 'buddies' to encourage matchers to ask questions.

6.0 General Usability

6.1 Findings

- Typically, CM were confident on how to use the system and understood the terms used on the buttons. However, there were instances where usability issues caused frustration to the matcher or decreased their speed (as outlined previously in section 4.1 Qualitative Findings).
- It was noted that a common error (possible coding error) appeared to arise with marital status, for example in one record it would say the individual was widowed and in the other say the individual was in a civil partnership. It varied whether matchers sent these to expert or decided to match based on the other available information.
- CM frequently used the 'more information' and 'household' buttons to aid their decision making. They understood the difference between the two buttons; more information to show more details on the individual in a record and household to show relationships.
- Typically, matchers used the 'send to expert' feature when they couldn't be certain on a match. It varied how much information they wrote within the write-in box available; some noted they ***"need more info"*** whilst others would provide an explanation for what information they would like i.e. scanned image and why they considered it a possible match.
- There were instances when the 'send to expert' and 'report link' were used interchangeably for the same queries, for example when there was missing name or date of birth, it varied whether matchers would send this to expert or use the report link.
- The 'send to expert' feature was used more frequently than the 'report link'. It was noted that they would send to expert for the same query multiple times within a day, so would prefer an option to get their question answered instantly, to avoid having to send future clusters to an expert.
- The dashboard was found to be easy to navigate and fit for purpose. Matchers understood what to expect upon pressing 'view' under each form of matching. However, matchers did not always understand what 'pre search' is, despite receiving training on this. There were suggestions that 'target matching' would be a more appropriate term.

6.2 Recommendations

- Consider implementing a system to allow matchers to ask questions straight away i.e. buddy or open skype call, to avoid sending the same query to expert throughout the day/ week.

- Clarify in training when matchers should use 'send to expert' feature vs. 'report link' and check their understanding of this.

7.0 Individual matching

7.1 Findings

- CM interacted with the system accurately and found it straightforward.
- They were slower in pace within the first 10 minutes, but this quickly increased, and a good pace was maintained thereafter.
- They used the 'household' page more frequently than the 'more information' page, as they said the 'more information' page had more missing information.
- They did not use the sort by function as they said it was not needed, but understood it was available and how to interact with it.
- CM understood what 'same person' and 'no more matches' meant and used these buttons correctly and consistently based on whether they thought records were a match or not.
- There was some confusion initially about whether it was possible to select more than one match, but this was quickly remembered by the matcher when they recalled the training.
- It was noted that matchers would have liked to know what to do if there were blanks in the information. Typically, in these instances they would send to expert or make decisions based on the available information, as opposed to selecting 'report issue', as this did not appear to be available on the 'individual matching' prototype.
- They rarely used the 'send to expert' feature, in instances when it was felt they should have. When they did use the feature, they did not provide much detail in the write in field and moved onto the next cluster quickly.

7.2 Recommendations

- Consider including an active 'report issue' link on this page, to keep consistency.
- Provide further training on what action to take when there is missing information i.e. when to send to expert and how much information should be recorded in the write in.

8.0 Household matching

8.1 Findings

- CM consistently used the 'more information' and 'household' pages in order to make their decisions.
- It varied whether CM completed household matching correctly;
    - Some matchers understood the purpose of household matching, how to match the records and followed the process correctly i.e. make decision on page 1, move on to match records on page 2 then back to page 1 to review and submit. In these instances, CM did not tick the people in the household, unless they wished to 'break person match'.
    - There were also instances where CM did not follow the advised process. They would review details on page 1 but not make any decisions, before making decisions on page 2, then go back to page 1 to make a decision (ticking all people within household and both households and selecting 'same address') and then pressing 'submit'. There were also instances of interacting with page 2 of the household system inaccurately; they did not understand the purpose was to match unmatched records, but instead thought they had to add any potential household members to the household on page 1, irrespective of if there was a match or not. For example, if they believed John Smith was part of household A, even if he was the only record on page 2, they would select 'same person' and add them to the household. If there were sibling records on this page (none of which they thought were matches to each other) they would try add them all to the household, which would create duplicate records – once they realised this they would 'break person match' and 'send to expert' on page 2. This meant they rarely used the 'no more matches' button.

        - In these instances, CM misunderstood ‘same person’ to have two meanings; ‘this is a match’ but also ‘add to household’, when there was only one record showing.
        - They understood page 2 of household matching to be to review potential household members and decide whether to add them to the household, opposed to looking for matches.
- CM did not like the process of having to go between the pages and said it was not very intuitive, as they would sometimes have to go back and forth multiple times. It was noted that it would be useful to stay on page 2 of the household interface until all matches had been made and the matcher had selected ‘no more matches’.
- There was a desire for more information on occupation and marital status, in order to make accurate decisions.
- Matchers typically had the highlighting feature on for this form of matching, however sometimes turned it off to aid their decision making, for example when the highlighting made it appear there were more differences than there were.
- It was not clear to matchers whether to tick household members when selecting ‘same address’ and ‘submit’, or whether the household members should only be ticked when selecting ‘break person match’.

8.2 Recommendations

- Provide more practical training for matchers on household matching, for example, sessions where they are observed by an expert completing matching to resolve any issues. Particular focus should be paid to the process of moving between the two pages and the purpose of page 2 of the household matching interface.
- Allow CM to make all their decisions on page 2 before automatically sending them back to page 1.

9.0 Pre search

9.1 Findings

- There were instances where CM had forgotten what ‘pre search’ meant, suggesting the name itself was not well understood. However, typically the type of matching was understood to be finding a match to a target record.
- Matchers interacted with this system accurately, consistently using the ‘household’ and ‘more information’ buttons to aid decision making.
- They used the sort by function often multiple times within one cluster, typically changing between sorting by name and date of birth depending on which information was available.
- Matchers reported it was easy and observations showed the system was used intuitively.
- It was noted that matchers would have liked to have been able to eliminate records, so that they don’t accidently view the same one multiple times. For example, if they were certain it wasn’t a match, they would like the option the make a record ***“greyed out”*** or disappear.
- No more information was felt to be needed to aid decisions, other than images. In these instances, the matchers would ‘send to expert’.
- The highlighting feature was found to be helpful for ‘pre search’ and was kept on consistently throughout.

9.2 Recommendations

- Provide an option for matchers to eliminate records they are certain are not a match.
- Clarify understanding of the term ‘pre search’ within training.