11-18-2022

The Development of a Mobile App-Focused Deduplication Strategy for the Apple Heart Study That Informs Recommendations for Future Digital Trials

Ariadna Garcia
Stanford University

Justin Lee
Stanford University

Vidhya Balasubramanian
Stanford University

Rebecca Gardner
Stanford University

Santosh E. Gummidipundi
Stanford University

See next page for additional authors

Authors

Ariadna Garcia, Justin Lee, Vidhya Balasubramanian, Rebecca Gardner, Santosh E. Gummidipundi, Grace Hung, Todd Ferris, Lauren Cheung, Sumbul Desai, Christopher B. Granger, Mellanie True Hills, Peter Kowey, Divya Nag, John S. Rumsfeld, Andrea M. Russo, Jeffrey W. Stein, Nisha Talati, David Tsay, Kenneth W. Mahaffey, Marco V. Perez, Mintu P. Turakhia, Haley Hedlin, and Manisha Desai

Received: 4 March 2022 | Revised: 28 April 2022 | Accepted: 30 April 2022

DOI: 10.1002/sta4.470

SPECIAL ISSUE ARTICLE

WILEY

The development of a mobile app-focused deduplication strategy for the Apple Heart Study that informs recommendations for future digital trials

Ariadna Garcia[1,2] | Justin Lee[1,2] | Vidhya Balasubramanian[1,2] | Rebecca Gardner[1,2] | Santosh E. Gummidipundi[1,2] | Grace Hung[3] | Todd Ferris[3] | Lauren Cheung[4] | Sumbul Desai[4] | Christopher B. Granger[5] | Mellanie True Hills[6] | Peter Kowey[7] | Divya Nag[4] | John S. Rumsfeld[8] | Andrea M. Russo[9] | Jeffrey W. Stein[4] | Nisha Talati[10] | David Tsay[4] | Kenneth W. Mahaffey[10] | Marco V. Perez[2] | Mintu P. Turakhia[2,11] | Haley Hedlin[1,2] | Manisha Desai[1,2] | on behalf of the Apple Heart Study Investigators

[1]Quantitative Sciences Unit, Stanford University School of Medicine, Stanford, California, USA

[2]Department of Medicine, Stanford University School of Medicine, Stanford, California, USA

[3]Technology and Digital Solutions, Stanford Health Care and School of Medicine, California, Stanford, USA

[4]Apple Inc., Cupertino, California, USA

[5]Duke Clinical Research Institute, Duke University, Durham, North Carolina, USA

[6]StopAfib.org, American Foundation for Women's Health, Decatur, Texas, USA

[7]Lankenau Heart Institute and Jefferson Medical College, Philadelphia, Pennsylvania, USA

[8]Department of Medicine, University of Colorado School of Medicine, Aurora, Colorado, USA

[9]Department of Medicine, Cooper Medical School of Rowan University, Camden, New Jersey, USA

[10]Stanford Center for Clinical Research, Stanford University School of Medicine, Stanford, California, USA

[11]Center for Digital Health, Stanford University School of Medicine, Stanford, California, USA

Correspondence
Manisha Desai, Quantitative Sciences Unit, Stanford University, 1701 Page Mill Road, Palo Alto, CA 94304, USA.
Email: manishad@stanford.edu

Funding information
This work was partially supported by Apple Inc. for work conducted for the Apple Heart Study, and partially supported by National Institute of Health Grants UL1TR003142 and P30CA124435.

Abstract

An app-based clinical trial enrolment process can contribute to duplicated records, carrying data management implications. Our objective was to identify duplicated records in real time in the Apple Heart Study (AHS). We leveraged personal identifiable information (PII) to develop a dissimilarity score (DS) using the Damerau–Levenshtein distance. For computational efficiency, we focused on four types of records at the highest risk of duplication. We used the receiver operating curve (ROC) and resampling methods to derive and validate a decision rule to classify duplicated records. We identified 16,398 (4%) duplicated participants, resulting in 419,297 unique participants out of a total of 438,435 possible. Our decision rule yielded a high positive predictive value (96%) with negligible impact on the trial's original findings. Our findings provide principled solutions for future digital trials. When establishing deduplication procedures for digital trials, we recommend collecting device identifiers in addition to participant identifiers; collecting and ensuring secure access to PII; conducting a pilot study to identify reasons for duplicated records; establishing an initial deduplication algorithm that can be refined; creating a data quality plan that informs refinement; and embedding the initial deduplication algorithm in the enrolment platform to ensure unique enrolment and linkage to previous records.

KEYWORDS
Damerau-Levenshtein distance, deduplication, digital trial, entity resolution, pragmatic trial, resampling methods

1 | INTRODUCTION

Over the past few years, there has been increased interest in conducting pragmatic clinical trials (PCTs). PCTs are cost effective; they can reduce the burden on participants and the study team by providing greater flexibility on where, when and how the data are collected. The digital clinical trial (DCT) is a special case of the PCT that further allows investigators to incorporate mobile devices or digital tools to facilitate the conduct of a trial and/or to evaluate the use of digital platforms in an intervention in a real-life setting (Inan et al., 2020). Mobile apps and wearable devices enable DCTs to provide opportunities to engage a larger number of participants with many more data elements collected per individual. Although it may be advantageous to have many more measurements of a given type and to have a diversity of types of measurements to address the study objectives, management of the data may become challenging and may pose threats to the integrity of the trial.

For example, studies that rely on apps for enrolment are at an increased risk of duplicated data, which can have serious implications on downstream analyses as well as issues for study conduct. In a systematic review conducted by Zhang et al. (2018), the authors state that 'scholars have only started to employ apps in field experiments in the last 4 years' and 'most studies only used apps as an experimental treatment instead of an experimental platform'. Though many have studied this issue extensively by providing solutions in other contexts (Chaudhuri et al., 2003; Gravano et al., 2001), the challenges faced when conducting PCTs and mitigation solutions have yet to be fully characterized. Further, the specific issue of duplication has not been well described in the DCT literature. For example, the Clinical Trials Transformation Initiative (2022) provides an excellent set of guidelines to address various issues in the conduct, design and analysis of clinical trials that rely on digital tools. This particular issue, however, is not featured as one for investigators to consider in the design and analysis. Thus, currently, there is a gap in knowledge of how to plan for and mitigate deduplication issues when conducting DCTs.

The Apple Heart Study (AHS) was a prospective, single-arm, site-less, pragmatic study (Turakhia et al., 2019) conducted between 11/29/2017 and 02/21/2019, which faced and overcame some of these data management challenges. More specifically, the goal of the AHS was to evaluate the ability of an irregular pulse notification on the Apple Watch to identify signals consistent with atrial fibrillation. The study relied on an app on the participant's phone to enroll participants, collect data and monitor the heart rhythm for abnormalities using pulse data recorded on their Apple Watch. The AHS was a collaborative project between Apple Inc. and Stanford University. Our study team consisted of experts from both organizations that covered diverse disciplines including cardiovascular medicine, digital health, software engineering, biostatistics, epidemiology, clinical trials, data management, clinical informatics and clinical operations. Clinical operations were led by the Stanford Center for Clinical Research. Design, data management and data analysis were led by the Stanford Quantitative Sciences Unit. Data capture, security and housing were led by the Stanford Technology and Digital Solutions Team, and the Stanford Center for Digital Health provided expertise in the area of digital health. Key external collaborators included AmericanWell, a TeleHealth provider, and BioTelemetry, an ambulatory electrocardiogram provider. Governance of the AHS included an Executive Committee of strategic investigators that drove day-to-day decisions. The Executive Committee was advised by a larger Steering Committee of representatives from Stanford University, Apple Inc. and five external members including a patient advocate with atrial fibrillation. Additionally, a Data and Safety Monitoring Board was established to advise the team on trial integrity, conduct, safety and dissemination. Using team science principles, the study team met regularly (weekly meetings with the larger study team and biweekly meetings with smaller sub-teams) to troubleshoot issues on study design, launch, enrolment, data management, analysis and interpretation of findings.

Through data monitoring procedures, the study team observed early in the recruitment phase that some participants were represented under multiple participant identifiers or IDs—an occurrence that we refer to as duplication of IDs—making it challenging to identify the unique number of participants enrolled and to link longitudinal data within an individual. Both tasks were critical to accomplish study goals. Behaviours of the participants and/or the devices may have contributed to duplication of records. Some examples include participants deleting, reinstalling and re-enrolling through the app; app crashes or software updates that could cause loss of data; switching mobile devices; or sharing iCloud accounts with other individuals.

Entity resolution (ER) is the task of identifying and grouping various manifestations of the same real-world object within one data source or between different sources (Benjelloun et al., 2009; Köpcke et al., 2010). This area of research is also referred to as deduplication, record linkage, object matching or linkage discovery. ER has several applications particularly in web searches, health data, financial transactions, law enforcement and more.

In this paper, we describe an issue with ER specifically as it arose in the AHS. We describe our approach to mitigating this problem so that we could achieve study goals. Lessons learned are translated into recommendations for future DCTs.

2 | METHODS

2.1 | Background

When designing the AHS (Turakhia et al., 2019), we established a system to verify unique identifiers that addressed anticipated scenarios that could lead to duplication issues. Our system involved having an ID for the participant (participant ID or PID) and an additional ID for the device (device ID or DID) so that data generated from a device would have both a corresponding PID and DID linked to a single participant. Data not generated from the device (e.g. study visit) would only have the PID. In the following, we describe in greater detail these two types of identifiers.

2.2 | DID and PID

When a participant downloaded and opened the app, encrypted data were securely sent to Apple servers through an application program interface (API). The data included a system identifier referred to as the DID that allowed Apple servers and participant devices to exchange data. In addition, when a participant enrolled in the study through the app, they were assigned a PID generated by the app. Upon assignment of a PID to a participant, a mapping was established between the DID and PID.

We define duplication as the phenomenon of having multiple de-linked records that appeared to belong to the same unique individual. This can occur in multiple ways. A new DID could be assigned to the same device throughout the study for a variety of reasons including software updates, device reboot and device updates. Like DIDs, new PIDs may be assigned to a participant with an existing assigned PID for a variety of reasons (Figure 1). Duplication (multiple PIDs corresponding to the same individual) occurs when either or both PID and DID are reassigned to the same person or device leading the data to become orphaned or no longer linked, where pieces of data from the same participant are mistakenly thought to belong to separate individuals. To solve these issues, we used ER techniques and developed an algorithm to establish which PIDs were associated with the same individual. The algorithm was designed to link data from multiple sources back together and create a true unique identifier per person enabling longitudinal views of data within a participant over time. The algorithm consists of multiple steps described in detail that follow and depicted in detail in Figure 2.

2.3 | Key inputs to the algorithm

The algorithm leveraged data that contained participant identifiable information (PII), obtained by asking participants questions related to their identity and demographic information after providing consent. These data were encrypted, not accessible to the sponsor (Apple Inc.), and only available to an unblinded sub-team of the larger AHS data science team at Stanford University.

The algorithm specifically used the combination of seven participant-level identifiers: email, date of birth, first name, last name, phone number, state and consent date in order to assess the similarity of two records with different PIDs. All identifiers provided important information to

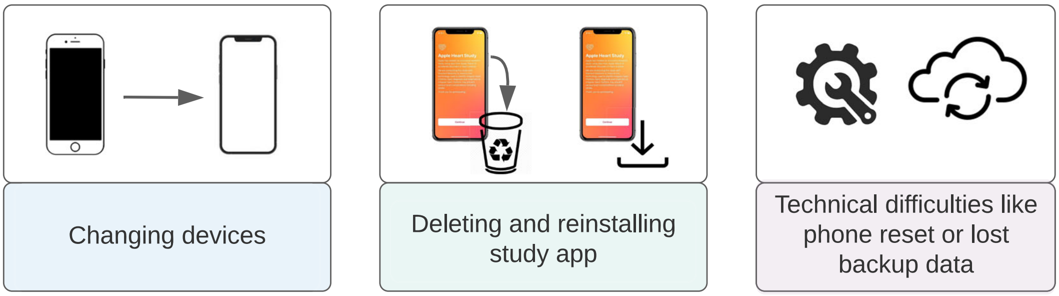
Changing devices
Deleting and reinstalling study app
Technical difficulties like phone reset or lost backup data

FIGURE 1 Common scenarios that potentially led to orphaned data

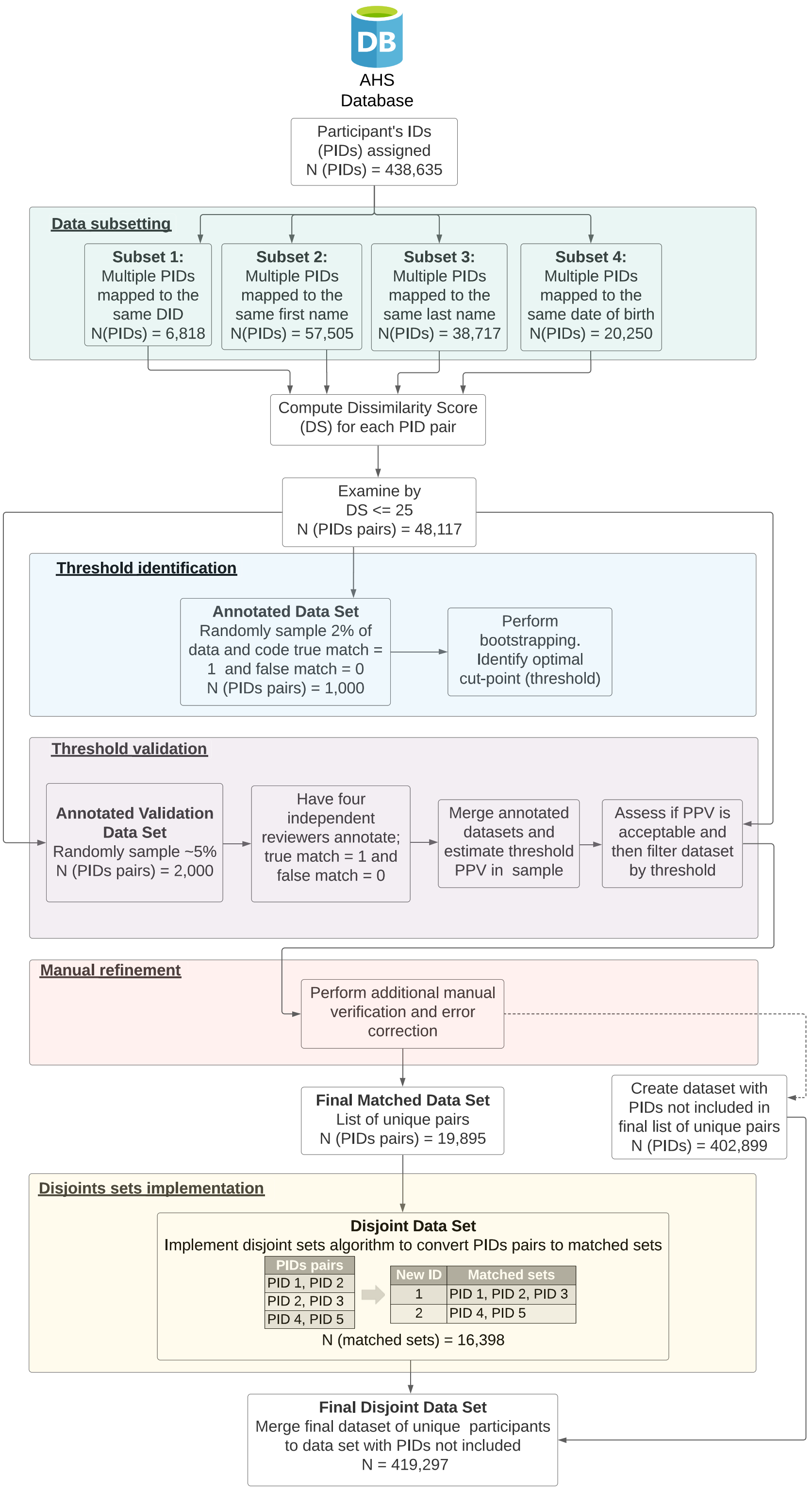
DB
AHS Database
Participant's IDs (PIDs) assigned N (PIDs) = 438,635
Data subsetting
Subset 1: Multiple PIDs mapped to the same DID N(PIDs) = 6,818
Subset 2: Multiple PIDs mapped to the same first name N(PIDs) = 57,505
Subset 3: Multiple PIDs mapped to the same last name N(PIDs) = 38,717
Subset 4: Multiple PIDs mapped to the same date of birth N(PIDs) = 20,250
Compute Dissimilarity Score (DS) for each PID pair
Examine by DS <= 25 N (PIDs pairs) = 48,117
Threshold identification
Annotated Data Set Randomly sample 2% of data and code true match = 1 and false match = 0 N (PIDs pairs) = 1,000
Perform bootstrapping. Identify optimal cut-point (threshold)
Threshold validation
Annotated Validation Data Set Randomly sample ~5% N (PIDs pairs) = 2,000
Have four independent reviewers annotate; true match = 1 and false match = 0
Merge annotated datasets and estimate threshold PPV in sample
Assess if PPV is acceptable and then filter dataset by threshold
Manual refinement
Perform additional manual verification and error correction
Final Matched Data Set List of unique pairs N (PIDs pairs) = 19,895
Create dataset with PIDs not included in final list of unique pairs N (PIDs) = 402,899
Disjoints sets implementation
Disjoint Data Set
Implement disjoint sets algorithm to convert PIDs pairs to matched sets
PIDs pairs
PID 1, PID 2
PID 2, PID 3
PID 4, PID 5
New ID
Matched sets
1
PID 1, PID 2, PID 3
2
PID 4, PID 5
N (matched sets) = 16,398
Final Disjoint Data Set Merge final dataset of unique participants to data set with PIDs not included N = 419,297

FIGURE 2 Step-by-step diagram of deduplication algorithm. DID, Device ID; DS, Dissimilarity score; PID, Participant's ID; PPV, positive predictive value

our algorithm with varying levels of contribution; email, date of birth, first name, last name and phone allowed us to identify the individual and were referred to as strong identifiers, whereas the others were considered auxiliary to the strong identifiers. For example, the participant's state was useful in scenarios where individuals shared their first and last names and birthdays but differed by state, casting doubt on the records being linked. Consent date was also helpful. For example, suppose an individual enrolled and then subsequently updated the app and re-enrolled. In this case two different PIDs would be attached to the respective data collected at the separate enrolments. Common PIIs with distinct consent dates close in time may reflect this sequence of events and increase the probability that the records belong to the same individual. Alternatively, common PIIs with the same consent date may indicate different family members joining the study simultaneously.

2.4 | Dissimilarity score (DS)

There are numerous algorithms that have been developed to calculate the distance between two string values (Cohen et al., 2003). We used the optimal string-alignment distance (osa) that relies on the Damerau–Levenshtein distance and returns the string distance taking into account deletion, insertion, substitution and transposition under the condition that no substring is edited more than once (Levenshtein, 1966). We implemented this using the stringdist R package (Van der Loo et al., 2014).

To calculate the dissimilarity score (DS) between two records, we first pre-processed the data by standardizing strings, removing special characters including empty spaces and transforming all text to the lower case. We then calculated approximate string distance between each of the patient-level identifiers and then summed the individual metrics to generate a single composite score that represented their dissimilarity. For example, email from one record was compared with email from a second record, and the string was quantified and recorded. The same process was repeated for each of the seven patient-level identifiers. A DS of zero indicated the two strings being compared were identical, whereas non-zero DSs reflected the degree of dissimilarity, that is, higher scores indicated greater dissimilarity.

2.5 | Algorithm execution and data sub-setting

One downside of performing pairwise string comparison for a high volume of records is that it is computational expensive. Assessing distance among all pairs in the full data would have required over 96 billion pairwise comparisons. To mitigate this challenge, we identified four subsets of data where true matches were most likely to occur. The first subset was composed of record pairs with multiple PIDs associated with the same DID (Subset 1: multiple PIDs mapped to the same DID). The second and third subsets were record pairs with multiple PIDs associated with the same first and last name, respectively (Subset 2: multiple PIDs mapped to the same first name; Subset 3: multiple PIDs mapped to the same last name), and the fourth was formed by participants where multiple PIDs were associated with the same date of birth (Subset 4: multiple PIDs mapped to the same date of birth). The DS was calculated for each pairing in each of these four subsets.

2.6 | Threshold identification

Empirical assessments of comparisons demonstrated that DSs > = 25 clearly corresponded to records from distinct participants, whereas comparisons with DSs < 25 more likely reflected a mixture of the same and different individuals. Thus, we restricted our focus on deriving a rule for records with DSs < 25.

For this purpose, we created an Annotated Data Set by randomly sampling 2% of pairwise comparisons with DSs < 25 and annotating each pair using a single reviewer to indicate whether the match was false (denoted as 0) or true (denoted as 1).

To identify the optimal threshold, we bootstrapped 10,000 samples from the Annotated Data Set, where within each bootstrap sample, a random 90% sample of the Annotated Data Set was allocated to a training data set in which the optimal cut-point was calculated using the Youden index (Youden, 1950) that minimizes misclassification. This cut-point was then applied to the remaining 10% of the bootstrap sample to estimate the accuracy for this specific cut-point. These results were summarized by evaluating the mean, median, mode and spread and we chose the cut-point as the mean of the 10,000 samples. The R package 'OptimalCutpoints', which defines the optimal cut as the point that maximizes the Youden function (the difference between true and false positive rates over all possible cut-point values) using an empirical approach, was used to execute approaches (Lopez-Raton & Rodriguez-Alvarez, 2021).

2.7 | Threshold validation

To assess the positive predictive value of the threshold in identifying true matches, we randomly resampled 5% of the original data set (excluding the 2% sample previously used) from records that had DSs under the identified threshold for annotation performed by four reviewers and called this new data set the Annotated *Validation* Data Set. There was no formal training for the reviewers or protocol for annotation, and subjective judgement was used. The final annotated value was taken to be the one defined by the majority of the four reviewers. In cases where majority could not be achieved, there was collaborative discussion to arrive at consensus. The positive predictive value was estimated as the proportion of the records in the Annotated Validation Data Set that was truly determined to be a duplicated record as defined by the final annotated value.

2.8 | Manual refinement

Minimizing match errors was critical to our study, so after completing the algorithm execution, we performed a manual verification process where we manually examined pairs that were slightly above or below the threshold. Erroneous instances were corrected manually. Once we completed this manual verification and correction process, the pairs identified as true matches comprised our final matched data set.

2.9 | Disjoint sets implementation

The last step was to find the intersections among all the matched pairs (referred to as the Final Matched Data Set). For example, suppose that PID 1 matched to PID 2 and PID 2 matched to PID 3. We then established that PID 1, PID 2 and PID 3 all corresponded to the same individual.

To understand the connection of all PIDs based on the pairwise matches, we first identified disjoints sets (pairs of matched PIDs with no PID in common). Those in the disjoint sets resulted in one unique ID per pair. PIDs that were not included in the disjoint data set were (1) those not included in one of the original four subsets with higher probability of being duplicates and (2) those PIDs excluded during the final matched data set creation (either by being above the threshold or by being deemed 'not a match' during the manual refinement step).

After executing the disjoint sets exercise described previously, we were able to create a data set of disjoint sets (Disjoint Data Set) where each row contained all PIDs that belonged together. Using the example above, suppose that the following matched pairs existed on the Final Matched Data Set: PID 1 matched to PID 2, PID 2 matched to PID 3, and PID 4 matched to PID 5. In this scenario, PID 1, PID 2 and PID 3 corresponded to the same individual, and PID 4 and PID 5 belonged to a separate individual because there were no PIDs in common. From this data set, we can arrive at unique IDs for those with multiple PIDs (Final Disjoint Data Set). A new unique ID was then added to each row allowing us to link PIDs that mapped to the same unique identifier and establish the longitudinal trajectory of each participant.

2.9.1 | Assessing inter-annotation reliability

To assess inter-rater reliability among the four reviewers of the Annotated Validation Data Set used to describe the properties of the estimated cut-point, each pair was recoded by the other reviewers, and Fleiss's kappa was calculated (Fleiss et al., 2003).

3 | RESULTS

3.1 | Algorithm execution and data sub-setting

AHS had a total of 438,435 PIDs assigned in the study cohort by the end of the enrollment period (Figure 2). Recall from above that to make the algorithm computationally efficient, we created four subsets where two or more PIDs were associated with the same piece of data. Subset 1 (Multiple PIDs mapped to the same DID) was composed of 6818 unique PIDs where two or more PIDs were associated with the same DID. Subsets 2 and 3 (multiple PIDs mapped to the same first and last name, respectively)—the most common records at risk of duplication that included participants—were composed of 57,505 and 38,717 PIDs. The last subset, Subset 4 (multiple PIDs mapped to the same date of birth), was composed of 20,250 PIDs.

3.2 | Threshold identification

DS was computed for each pair of PIDs, where DS was the sum of each of the seven individual identifiers. A description of the DS distribution and the number of pairs can be found in Table 1. The mean DS on the subsets where PIDs were associated with the same DID (mean = 8) and the date of birth (mean = 9) was lower than in the subsets where PIDs were associated with the same first name (mean = 16) or last name (mean = 14). This was expected as first and last names alone are weaker identifiers than DID or the date of birth.

PID pairs with DS < = 25 were then merged into a single data set, yielding a total of 48,117 unique PID pairs. A sample of approximately 2% was then extracted from this set of pairs ($N = 1000$, Min DS = 3, Max DS = 25) and manually coded to indicate true matches versus false matches.

We bootstrapped 10,000 times leaving 10% of the observations out each time in order to estimate and evaluate the optimal cut-point. After summarizing resampling findings, we found the optimal cut-point to be a DS of 22 (mean, median and mode of cut-points across samples were all equivalent and equal to 22) with sensitivity = 0.986, 1 − specificity = 0.009 and AUC = 0.999 as shown in Figure 3. Note, however, that these properties are presented on the same data set used to identify the threshold and are therefore an overly optimistic view of the rule's properties.

TABLE 1 Dissimilarity score (DS) characteristics by subset

Subset	
Subset 1: Multiple PIDs mapped to same DID	***N***
N (PIDs)	6818
N (pairs with DS < = 25)	2715
DS distribution	
Min	0
Median	7
Mean	8
Max	25
Subset 2: Multiple PIDs mapped to same first name	
N (PIDs)	57,505
N (pairs with DS < = 25)	36,884
DS distribution	
Min	0
Median	18
Mean	16
Max	25
Subset 3: Multiple PIDs mapped to same last name	
N (PIDs)	38,717
N (pairs with DS < = 25)	30,395
DS distribution	
Min	0
Median	9
Mean	14
Max	25
Subset 4: Multiple PIDs mapped to same date of birth	
N (PIDs)	20,250
N (pairs with DS < = 25)	20,063
DS distribution	
Min	0
Median	8
Mean	9
Max	25

3.3 | Threshold validation

Using the Annotation Validation Data Set of 2000 observations, we found that a DS < = 22 provided a positive predictive value of 96% with the 95% confidence interval that spanned from 95% to 97%. All PID pairs were then filtered to include only pairs with DS < = 22. Based on the observed results we then restricted our Final Matched Data Set to those pairs with a DS < = 22.

3.4 | Manual refinement

We performed additional manual validation of the data to ensure that those that had met the threshold were indeed a correct match and those that had been left out but were closer to the threshold were not actual matches that had been left out accidentally. If erroneous instances were identified, we then corrected that instance manually. Of the 1500 records that were manually evaluated, we encountered between 10 and 15 instances where manual correction was necessary (as this was an ad hoc procedure, we did not record the actual number of instances and are relying on memory). The resulting Final Matched Data Set included a total of 19,895 unique PID pairs.

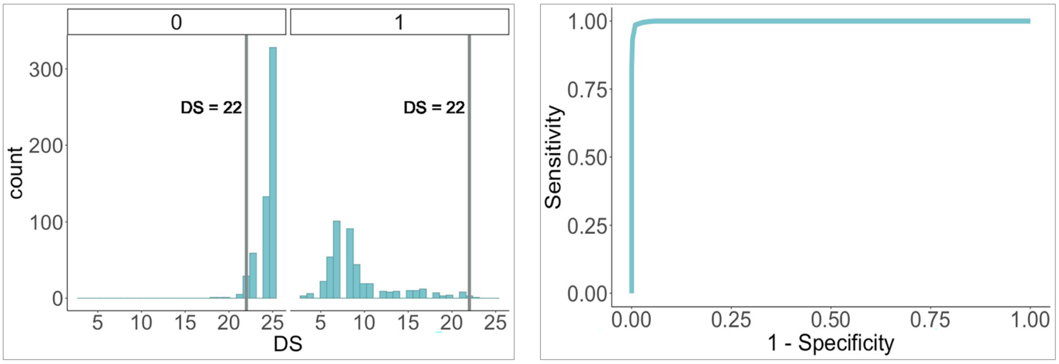

FIGURE 3 Left panel: Sample distribution by class (false match = 0 and true match = 1). Right panel: ROC for optimal cut-point

3.5 | Disjoint sets implementation

Application of the disjoint set algorithm resulted in 16,398 disjoint sets (or unique individuals). We found that on average, the Final Matched Data Set had two PIDs associated with the same individual and the number of PIDs associated with the same person ranged from 2 to 34.

All the PIDs that were not included in the Final Disjoint Data Set of 16,398 individuals were considered unique and not duplicated participants ($N = 402{,}899$). A new ID was assigned to each row of the merged data set, yielding our final cohort denominator of 419,297 participants.

3.6 | Assessing inter-annotation reliability

The interrater reliability of the four reviewers who independently coded the validation data set had a corresponding kappa statistic of 0.87, 95% CI (0.82, 0.91).

4 | DISCUSSION

Despite having an initial system for identifying duplicated records, we still encountered duplicated participants in the AHS and were unexpectedly faced with having to refine a deduplication algorithm while the study was ongoing to ensure data quality and the integrity of study conduct and research findings. Our algorithm relied on the Damerau–Levenshtein distance to create a composite score to quantify the dissimilarity of two given records and employed resampling techniques to estimate and validate a decision rule. We found that of all PIDs generated, approximately 4% were duplicated. To increase computational efficiency, our algorithm focused on subsets of data where participants shared some information in common; this strategy along with parallelization allowed us to make our algorithm more computational efficient, reducing the number of comparisons from 96 billion to less than 3 billion. Our algorithm allowed us to map each PID with high accuracy to orphaned data with negligible impact to the original trial findings. Specifically, we found that our algorithm had a positive predictive value of 96% (with 95% confidence interval spanning from 95% to 97%). For the original goals of the AHS study, this means that even if the true positive predictive value were only 95% (the lower limit of the confidence interval), we would have falsely classified approximately 5% of the 16,398 cases or 819 records as being duplicates when they were not. The implications are that our estimated sample size of enrolled participants would go from 419,297 to 422,856. When calculating our main estimate—the irregular pulse notification rate—this would have changed the estimate from 0.52% to 0.51% (Perez et al., 2019). Thus, for the purposes of the AHS, the properties of the algorithm are such that potential changes to the interpretation of trial findings were minimal. For data sets of varying sizes, an understanding of the implications of an algorithm with a similar positive predictive value may vary depending on the sizes of the targeted enrollment and the Annotated Validation Data Set. For example, suppose application of the algorithm to a data set of 200 records similarly results in 4% of the records identified as duplicated. Suppose further that an Annotated Validation Data Set of 50 is established to characterize the positive predictive value. For an algorithm with similar performance (the positive predictive value of 96%), the lower bound of the confidence interval for the positive predictive value is 91%, meaning that one of the eight records may have been falsely flagged so that the estimated sample size of 192 may be actually as low as 191.

There are strengths to our approach. We included multiple reviewers in our validation step to minimize bias in our validation data set. We had access to PII data that allowed us to leverage ER techniques to deduplicate data and reduce the burden of looking at thousands of participants manually to identify duplicate participants. Adding manual validation and spot-checking after execution of the algorithm was advantageous. The human brain can capture context in a way that algorithms may not. For example, in instances where we had two family members with nearly identical

TABLE 2 Hypothetical example of family related match

PID1	PID2	e-mail 1	e-mail 2	Date of birth 1	Date of birth 2	First name 1	First name 2	Last name 1	Last name 2	Phone number 1	Phone number 2	State 1	State 2	Consent 1	Consent2
1	2	eeboy55@email.Com	eegirl55@email.Com	1/21/77	5/24/81	Mike	Barbara	Thomas	Thomas	1–111–111-1111	1–111–111-1111	IN	IN	6/22/18 7:30	6/22/18 7:13

TABLE 3 Hypothetical example of common name match

PID1	PID2	e-mail 1	e-mail 2	Date of birth 1	Date of birth 2	First name 1	First name 2	Last name 1	Last name 2	Phone number 1	Phone number 2	State 1	State 2	Consent 1	Consent2
3	4	eemail.email59@mac.com	eemail.email1@mac.com	9/6/59	7/18/59	Tim	Tim	Thomas	Thomas	555–555-5555	655–566-6666	OK	CA	5/7/18 5:36	5/7/18 6:40

identifiers, differences in the records became immediately obvious to a reviewer who would notice that the two individuals were related but not the same. To illustrate this concept, Table 2 shows two participants with a DS of 18 (below the 22 threshold suggesting a match). At a closer look, we can conclude that these participants might be related because while they share the same phone number and last name, their first names are quite different (one is not a nickname of the other) and the pattern of their email naming system is similar although it results in distinct emails. Thus, the two IDs likely correspond to two different individuals from the same household. Another example is shown in Table 3, where common names accompanied by some similarities in identifiers (similar emails, year of birth and phone number) can suggest similarity by the algorithm but that are easily detected as distinct by the human brain. Finally, the principles behind our algorithm may be applied to data sets of varying sizes. For ease of adoption of the methods described here, we provide a simulated example data set of 10 records to illustrate how the code can be used (Appendix A).

There were also limitations to our study. We did not explore or compare other string distance methods. Additionally, our string distance relied on the summation of dissimilarity on a number of string items, and we did not consider other ways of constructing the DS. We also did not compare the performance of an algorithm that employs fewer identifiers or uses a combination of identifiers. In addition, though our bootstrapping procedure captures the uncertainty of our algorithm in identifying an optimal cut-point, we relied on the empirical approach for estimating Youden's index to identify an optimal cut-point. It may be that other methods for estimating Youden's cut-point are superior (Fluss et al., 2005). This is considered future work. Furthermore, we did not train reviewers when annotating in how to decide whether two given individuals were the same person (true match) or not (false match). Instead, we asked them to look at context and follow their intuition. Partly this was because we were unsure what to expect. This may have contributed to our agreement being less than ideal (kappa < 0.95). Finally, although our algorithm is designed to detect when multiple records come from the same individual, it does not address whether data classified under the same PID and DID may be arising from multiple individuals. Consider the scenario where multiple family members use the same device under one enrollment instance (one PID). Our algorithm will not detect such nuances.

There were important lessons that we gathered from our experience. Our study shows the importance of having deduplication algorithms in place prior to study launch. To that end, the collection of device IDs in addition to a participant ID on each piece of data is critical. Furthermore, the inclusion and secure access to PII data for deduplication algorithms are crucial to data scientists for ensuring data quality. Investigators should expect that the initially established deduplication algorithm will need to be further refined in an ongoing fashion. For this purpose, having a data quality plan in place that targets duplication is key. A pilot study can expose some of these challenges and should be conducted prior to launch. Finally, we highly recommend a plan embedded in the app to verify new enrollments at the time of enrollment via the initial deduplication algorithm. The potential participants can be prompted, for example, if their data 'match' a previous record while onboarding the study. Such preventive measures can go a long way to minimizing this issue from appearing downstream.

5 | CONCLUSIONS

Our experience in the AHS with the creation and implementation of a deduplication algorithm developed and validated in real time provides the foundation for six principles that can be borrowed when conducting other studies with similar pragmatic features. They include the following: Collect device IDs in addition to participant IDs; collect and ensure secure access to PII; conduct a pilot study to identify reasons for duplicated records; establish an initial deduplication algorithm that can be refined; create a data quality plan that informs refinement; embed the initial deduplication algorithm in the enrolment platform to ensure unique enrolment and linkage to previous records.

ETHICS STATEMENT

The research protocol was approved by the Institutional Review Board at Stanford University and by a central Institutional Review Board (Advarra). The research protocol including consenting procedures was approved by the Institutional Review Board at Stanford University and by a central Institutional Review Board (Advarra). No material has been reproduced from other sources. The work presented here was developed as part of the Apple Heart Study, which has the following ClinicalTrials.gov number, NCT03335800.

DATA AVAILABILITY STATEMENT

We are not able to provide data used to produce these results. We can, however, share code upon request.

ORCID

Rebecca Gardner https://orcid.org/0000-0001-5292-3219
Mellanie True Hills https://orcid.org/0000-0001-8298-4418
Manisha Desai https://orcid.org/0000-0002-6949-2651

REFERENCES

Benjelloun, O., Garcia-Molina, H., Menestrina, D., Su, Q., Whang, S. E., & Widom, J. (2009). Swoosh: A generic approach to entity resolution. *The VLDB Journal*, *18*(1), 255–276. https://doi.org/10.1007/s00778-008-0098-x

Chaudhuri, S., Ganjam, K., Ganti, V., & Motwani, R. (2003). Robust and efficient fuzzy match for online data cleaning. In Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data 2003 Jun 9 (pp. 313–324).

Clinical Trials Transformation Initiative. (2022). Digital health trials. Accessed on Feb 1, 2022, from https://ctti-clinicaltrials.org/our-work/digital-health-trials/

Cohen, W., Ravikumar, P., & Fienberg, S. (2003). A comparison of string metrics for matching names and records. In: Kdd workshop on data cleaning and object consolidation 2003 Aug 24 (Vol. 3, pp. 73–78).

Fleiss, J. L., Levin, B., & Paik, M. C. (2003). *Statistical methods for rates and proportions* (3rd ed.). John Wiley & Sons. https://doi.org/10.1002/0471445428

Fluss, R., Faraggi, D., & Reiser, B. (2005). Estimation of the Youden index and its associated cutoff point. *Biometrical Journal: Journal of Mathematical Methods in Biosciences*, *47*(4), 458–472. https://doi.org/10.1002/bimj.200410135

Gravano, L., Ipeirotis, P. G., Jagadish, H. V., Koudas, N., Muthukrishnan, S., & Srivastava, D. (2001). Approximate string joins in a database (almost) for free. In: VLDB 2001 11 (Vol. 1, pp. 491–500).

Inan, O. T., Tenaerts, P., Prindiville, S. A., Reynolds, H. R., Dizon, D. S., Cooper-Arnold, K., Turakhia, M., Pletcher, M. J., Preston, K. L., Krumholz, H. M., & Marlin, B. M. (2020). Digitizing clinical trials. *NPJ Digital Medicine*, *3*(1), 1–7. https://doi.org/10.1038/s41746-020-0302-y

Köpcke, H., Thor, A., & Rahm, E. (2010). Evaluation of entity resolution approaches on real-world match problems. *Proceedings of the VLDB Endowment*, *3*(1–2), 484–493. https://doi.org/10.14778/1920841.1920904

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, *10*(8), 707–710.

Lopez-Raton, M., & Rodriguez-Alvarez, M. X. (2021). Package 'OptimalCutpoints'. Accessed on Feb 1, 2022, from https://cran.r-project.org/web/packages/OptimalCutpoints/OptimalCutpoints.pdf

Perez, M. V., Mahaffey, K. M., Hedlin, H., Rumsfeld, J. S., Garcia, A., Ferris, T., Balasubramanian, V., Russo, A. M., Rajmane, A., Cheung, L., Hung, G., Lee, J. L., Kowey, P., Talati, N., Nag, D., Gummidipundi, S. E., Beatty, A., Hill, M., Desai, S., … Turakhia, M. P. (2019). Large-scale assessment of a smartwatch to identify atrial fibrillation. *New England Journal of Medicine*, *381*(20), 1909–1917. https://doi.org/10.1056/NEJMoa1901183

Turakhia, M. P., Desai, M., Hedlin, H., Rajmane, A., Talati, N., Ferris, T., Desai, S., Nag, D., Patel, M., Kowey, P., Rumsfeld, J. S., Russo, A. M., Hills, M. T., Granger, C. B., Mahaffey, K. W., & Perez, M. V. (2019). Rationale and design of a large-scale, app-based study to identify cardiac arrhythmias using a smartwatch: The apple heart study. *American Heart Journal*, *207*, 66–75. https://doi.org/10.1016/j.ahj.2018.09.002

Van der Loo, M., van der Laan, J., Logan, N., Muir, C., Gruber, J., & Ripley, B. (2014). The stringdist Package for Approximate String Matching. *The R Journal*.

Youden, W. J. (1950). Index for rating diagnostic tests. *Cancer*, *3*(1), 32–35. https://doi.org/10.1002/1097-0142(1950)3:1<32::aid-cncr2820030106>3.0.co;2-3

Zhang, J., Calabrese, C., Ding, J., Liu, M., & Zhang, B. (2018). Advantages and challenges in using mobile apps for field experiments: A systematic review and a case study. *Mobile Media & Communication*, *6*(2), 179–196. https://doi.org/10.1177/2050157917725550

How to cite this article: Garcia, A., Lee, J., Balasubramanian, V., Gardner, R., Gummidipundi, S. E., Hung, G., Ferris, T., Cheung, L., Desai, S., Granger, C. B., Hills, M. T., Kowey, P., Nag, D., Rumsfeld, J. S., Russo, A. M., Stein, J. W., Talati, N., Tsay, D., Mahaffey, K. W., … on behalf of the Apple Heart Study Investigators (2022). The development of a mobile app-focused deduplication strategy for the Apple Heart Study that informs recommendations for future digital trials. *Stat*, *11*(1), e470. https://doi.org/10.1002/sta4.470

APPENDIX A

This appendix provides a simulated data set, code and detailed annotation to walk the user through application of the deduplication algorithm.

R packages

```
library(dplyr)
library(tidyr)
library(tidyverse)
library(tibble)
library(stringr)
library(stringdist)
library(OptimalCutpoints)
library(reticulate) #-- This packages allows the user to execute Python code from Rmarkdown
```

Simulated data

Data containing duplicate individuals

```
data <- data.frame(
PID = c(1,2,3,4,5,6,7,8,9,10),
DID = c(1,2,3,4,5,6,3,4,2,3),
First_Name = c('Linda', 'Jennifer', 'Susan', 'Michael', 'James', 'Lidia', 'Sue', 'Maria','Jennifer','Sue'),
Last_Name = c('Smith', 'Williams', 'Brown', 'Jones', 'Davis', 'Smith', 'Brown', 'Jones', 'William','Brown'),
DOB = c('10/29/1964', '8/18/1965', '8/29/1972', '10/24/1972', '7/20/1988', '11/3/1993', '8/29/1972', '11/22/1972', '8/18/1966','8/29/1972'),
Email = c('lsmith22@email.com', 'jenniferwilliams@email.com', 'susanbrowng@email.com', 'mjones@email.com', 'jamesdavis44@email.com', 'lsmith12@email.com', 'susanbrowng@email.com', 'mjones@em
ail.com', 'jenniferwilliams@email.com', 'susanbrowng@email.com'),
State = c('MA','PA','TN','CA','TX','FL','TN','CA','PA','TN'),
Phone = c('453-245-0712','462-946-0095','630-512-5824','258-652-4875','880-391-9208','828-304-4350','630-512-5824','258-652-4875','462-946-0095','630-512-5825'),
Consent_Date = c('5/3/2017','5/10/2017','6/2/2017','6/29/2017','7/19/2017', '9/4/2017', '9/14/2017','10/20/2017','12/5/2017','10/24/17'))
```

PID	DID	First_Name	Last_Name	DOB	Email	State	Phone	Consent_Date
1	1	Linda	Smith	10/29/1964	lsmith22@email.com	MA	453-245-0712	5/3/2017
2	2	Jennifer	Williams	8/18/1965	jenniferwilliams@email.com	PA	462-946-0095	5/10/2017
3	3	Susan	Brown	8/29/1972	susanbrowng@email.com	TN	630-512-5824	6/2/2017
4	4	Michael	Jones	10/24/1972	mjones@email.com	CA	258-652-4875	6/29/2017
5	5	James	Davis	7/20/1988	jamesdavis44@email.com	TX	880-391-9208	7/19/2017
6	6	Lidia	Smith	11/3/1993	lsmith12@email.com	FL	828-304-4350	9/4/2017
7	3	Sue	Brown	8/29/1972	susanbrowng@email.com	TN	630-512-5824	9/14/2017
8	4	Maria	Jones	11/22/1972	mjones@email.com	CA	258-652-4875	10/20/2017
9	2	Jennifer	William	8/18/1966	jenniferwilliams@email.com	PA	462-946-0095	12/5/2017
10	3	Sue	Brown	8/18/1966S	susanbrowng@email.com	TN	630-512-5825	10/24/17

Data cleaning

Remove all non-alphanumeric characters (including spaces) and convert variables to the lower case. In our case we transformed: first name, last name, device ID (DID) and data of birth (DOB) but this can be extended to all the variables used in the matching function.

```
data$First_Name <- tolower(trimws(data$First_Name, which = "both"))
data$First_Name <- str_replace_all(data$First_Name,"[^[:alnum:]]", "")
data$Last_Name <- tolower(trimws(data$Last_Name, which = "both"))
data$Last_Name <- str_replace_all(data$Last_Name,"[^[:alnum:]]", "")
data$DID <- tolower(trimws(data$DID, which = "both"))
data$DOB <- trimws(data$DOB, which = "both")
```

Data sub-setting

Subset 1 (Multiple PIDs mapped to the same DID):

```
#-- Note, the code needs as.data.frame() at the end to be formatted correctly in order to work in the matching function
s1 <- data %>%
  group_by(DID) %>%
  filter(n() > 1) %>%
  arrange(DID) %>%
  as.data.frame()
```

PID	DID	First_Name	Last_Name	DOB	Email	State	Phone	Consent_Date
2	2	jennifer	williams	8/18/1965	jenniferwilliams@email.com	PA	462-946-0095	5/10/2017
9	2	jennifer	william	8/18/1966	jenniferwilliams@email.com	PA	462-946-0095	12/5/2017
3	3	susan	brown	8/29/1972	susanbrowng@email.com	TN	630-512-5824	6/2/2017
7	3	sue	brown	8/29/1972	susanbrowng@email.com	TN	630-512-5824	9/14/2017
10	3	sue	brown	8/29/1972	susanbrowng@email.com	TN	630-512-5825	10/24/17
4	4	michael	jones	10/24/1972	mjones@email.com	CA	258-652-4875	6/29/2017
8	4	maria	jones	11/22/1972	mjones@email.com	CA	258-652-4875	10/20/2017

Subset 2 (Multiple PIDs mapped to the same first name):

```
s2 <- data %>%
  group_by(First_Name) %>%
  filter(n() > 1) %>%
  arrange(First_Name) %>%
  as.data.frame()
```

PID	DID	First_Name	Last_Name	DOB	Email	State	Phone	Consent_Date
2	2	jennifer	williams	8/18/1965	jenniferwilliams@email.com	PA	462-946-0095	5/10/2017
9	2	jennifer	william	8/18/1966	jenniferwilliams@email.com	PA	462-946-0095	12/5/2017
7	3	sue	brown	8/29/1972	susanbrowng@email.com	TN	630-512-5824	9/14/2017
10	3	sue	brown	8/29/1972	susanbrowng@email.com	TN	630-512-5825	10/24/17

Subset 3 (Multiple PIDs mapped to the same last name):

```
s3 <- data %>%
  group_by(Last_Name) %>%
  filter(n() > 1) %>%
  arrange(Last_Name) %>%
  as.data.frame()
```

PID	DID	First_Name	Last_Name	DOB	Email	State	Phone	Consent_Date
3	3	susan	brown	8/29/1972	susanbrowng@email.com	TN	630-512-5824	6/2/2017
7	3	sue	brown	8/29/1972	susanbrowng@email.com	TN	630-512-5824	9/14/2017
10	3	sue	brown	8/29/1972	susanbrowng@email.com	TN	630-512-5825	10/24/17
4	4	michael	jones	10/24/1972	mjones@email.com	CA	258-652-4875	6/29/2017
8	4	maria	jones	11/22/1972	mjones@email.com	CA	258-652-4875	10/20/2017
1	1	linda	smith	10/29/1964	lsmith22@email.com	MA	453-245-0712	5/3/2017
6	6	lidia	smith	11/3/1993	lsmith12@email.com	FL	828-304-4350	9/4/2017

Subset 4 (Multiple PIDs mapped to the same date of birth):

```
s4 <- data %>%
  group_by(DOB) %>%
  filter(n() > 1) %>%
  arrange(DOB) %>%
  as.data.frame()
```

PID	DID	First_Name	Last_Name	DOB	Email	State	Phone	Consent_Date
3	3	susan	brown	8/29/1972	susanbrowng@email.com	TN	630-512-5824	6/2/2017
7	3	sue	brown	8/29/1972	susanbrowng@email.com	TN	630-512-5824	9/14/2017
10	3	sue	brown	8/29/1972	susanbrowng@email.com	TN	630-512-5825	10/24/17

Matching function

The following function takes four inputs:

1. **data**: This could be either the full data set or a subset (like the ones created above) depending on the size of the original data.
2. **id**: The name of the ID variable. In our example, this is the PID.
3. **vars**: The list of variables used to calculate the dissimilarity score (DS).
4. **cutoff**: This is an arbitrary and initial cut-off determined by the user that may be refined. The idea is to have a cut-off for which DSs below the cut-off may be at risk of duplication. Initially, the user can apply the function to a small subset to refine where the focus should be (i.e. to filter out those highly divergent and dissimilar records).

```
find_matches <- function(data,id, vars, cutoff)
{
  #-- create empty data frame to collect matches
  cols = paste0("data.frame(",id,".1 = character(0),", id,".2 = character(0),ds = numeric(0)")
  for(v in vars){
    cols = paste0(cols,",",v,".1 = character(0),", v,".2 = character(0)")}
    cols = paste0(cols, ")")
  m <- eval(parse(text=cols))  #-- m is the output data frame
  var.names = colnames(m)

  system.time(
    #-- loop through the entire input dataset
    for(i in seq(1,length(data[,c(id)]),1)){
      #-- create dummy variables with the corresponding string distance for each pairwise comparison
      for(v in vars){
          data[,c(paste0(v,".m"))] <- stringdist(data[i,c(v)], data[,c(v)])
        }
      #-- add DS variable to dataframe which is the sum of all the individual dummy variables created above
      data[,c("ds")] <-0
      data[,c("ds")] <- rowSums(data[,c(as.numeric(length(data)-length(vars)):as.numeric(length(data)-1))])
      #-- create temporary df with only those that fall within the cutoff pre-specified
      temp <- data[-i,] #-- exclude pairwise comparison to itself
      temp <- temp[which(temp$ds<=cutoff),]

      if(length(temp[,c(id)]) > 0)    #-- if there are matches within the pre-specified cuttoff
      {  #-- loop through each match, if the combination is not found add them to the output dataframe (m)
        for(j in seq(1,length(temp[,c(id)]),1))
        {
           if(!((data[i,c(id)] %in% m[,c(paste0(id,".1"))] & temp[j,c(id)] %in% m[,c(paste0(id,".2"))]) |
                (data[i,c(id)] %in% m[,c(paste0(id,".2"))] & temp[j,c(id)] %in% m[,c(paste0(id,".1"))]) |
                data[i,c(id)] == temp[j,c(id)]))

           {# -- combination does not exist so it gets added to matched set
            vals = paste0("data.frame(",id,".1 = \"",data[i,c(id)],"\",", id,".2 = \"",temp[j,c(id)],"\", ds = ",temp[j,c("ds")])
            for(v in vars){
              vals = paste0(vals,",",v,".1 = \"",data[i,c(v)],"\",", v,".2 = \"", temp[j,c(v)],"\"")}
            vals = paste0(vals, ")")

            nr = eval(parse(text=vals))
            m = rbind(m,nr)
            colnames(m) = var.names
          }
        }
      }
    })
  return(m)
}

#----------------------------------
#-- Test function example
#----------------------------------
#d = s1
#vars = c("First_Name", "Last_Name", "DOB", "Email", "State", "Phone", "Consent_Date")
#id = "PID"
#cutoff = 50
#m1 <- find_matches(d, id, vars, cutoff) %>% arrange(ds)
```

Execute matching function on each set

Pre-specified cut-off = 40

Note: Parallelization in this section is encouraged for large data sets.

```
vars = c("First_Name", "Last_Name", "DOB", "Email", "State", "Phone", "Consent_Date")
id = "PID"
cutoff = 40

m1 <- find_matches(s1, id, vars, cutoff) %>% arrange(ds)
m2 <- find_matches(s2, id, vars, cutoff) %>% arrange(ds)
m3 <- find_matches(s3, id, vars, cutoff) %>% arrange(ds)
m4 <- find_matches(s4, id, vars, cutoff) %>% arrange(ds)
```

Resulting matched sets

Matched Set 1

PID.1	PID.2	ds	First_Name.1	First_Name.2	Last_Name.1	Last_Name.2	DOB.1	DOB.2	Email.1
2	9	6	jennifer	jennifer	williams	william	8/18/1965	8/18/1966	jenniferwilliams@email.com
3	7	6	susan	sue	brown	brown	8/29/1972	8/29/1972	susanbrowng@email.com
7	10	6	sue	sue	brown	brown	8/29/1972	8/29/1972	susanbrowng@email.com
3	10	9	susan	sue	brown	brown	8/29/1972	8/29/1972	susanbrowng@email.com
4	8	10	michael	maria	jones	jones	10/24/1972	11/22/1972	mjones@email.com
3	4	35	susan	michael	brown	jones	8/29/1972	10/24/1972	susanbrowng@email.com
3	8	36	susan	maria	brown	jones	8/29/1972	11/22/1972	susanbrowng@email.com
10	4	38	sue	michael	brown	jones	8/29/1972	10/24/1972	susanbrowng@email.com

PID.1	Email.2	State.1	State.2	Phone.1	Phone.2	Consent_Date.1	Consent_Date.2
2	jenniferwilliams@email.com	PA	PA	462-946-0095	462-946-0095	5/10/2017	12/5/2017
3	susanbrowng@email.com	TN	TN	630-512-5824	630-512-5824	6/2/2017	9/14/2017
7	susanbrowng@email.com	TN	TN	630-512-5824	630-512-5825	9/14/2017	10/24/17
3	susanbrowng@email.com	TN	TN	630-512-5824	630-512-5825	6/2/2017	10/24/17
4	mjones@email.com	CA	CA	258-652-4875	258-652-4875	6/29/2017	10/20/2017
3	mjones@email.com	TN	CA	630-512-5824	258-652-4875	6/2/2017	6/29/2017
3	mjones@email.com	TN	CA	630-512-5824	258-652-4875	6/2/2017	10/20/2017
10	mjones@email.com	TN	CA	630-512-5825	258-652-4875	10/24/17	6/29/2017

Matched Set 2

PID.1	PID.2	ds	First_Name.1	First_Name.2	Last_Name.1	Last_Name.2	DOB.1	DOB.2	Email.1
2	9	6	jennifer	jennifer	williams	william	8/18/1965	8/18/1966	jenniferwilliams@email.com
7	10	6	sue	sue	brown	brown	8/29/1972	8/29/1972	susanbrowng@email.com

PID.1	Email.2	State.1	State.2	Phone.1	Phone.2	Consent_Date.1	Consent_Date.2
2	jenniferwilliams@email.com	PA	PA	462-946-0095	462-946-0095	5/10/2017	12/5/2017
7	susanbrowng@email.com	TN	TN	630-512-5824	630-512-5825	9/14/2017	10/24/17

Matched Set 3

PID.1	PID.2	ds	First_Name.1	First_Name.2	Last_Name.1	Last_Name.2	DOB.1	DOB.2	Email.1
3	7	6	susan	sue	brown	brown	8/29/1972	8/29/1972	susanbrowng@email.com
7	10	6	sue	sue	brown	brown	8/29/1972	8/29/1972	susanbrowng@email.com
3	10	9	susan	sue	brown	brown	8/29/1972	8/29/1972	susanbrowng@email.com
4	8	10	michael	maria	jones	jones	10/24/1972	11/22/1972	mjones@email.com
1	6	22	linda	lidia	smith	smith	10/29/1964	11/3/1993	lsmith22@email.com
8	1	34	maria	linda	jones	smith	11/22/1972	10/29/1964	mjones@email.com
3	4	35	susan	michael	brown	jones	8/29/1972	10/24/1972	susanbrowng@email.com
3	8	36	susan	maria	brown	jones	8/29/1972	11/22/1972	susanbrowng@email.com
3	1	38	susan	linda	brown	smith	8/29/1972	10/29/1964	susanbrowng@email.com
10	4	38	sue	michael	brown	jones	8/29/1972	10/24/1972	susanbrowng@email.com
3	6	40	susan	lidia	brown	smith	8/29/1972	11/3/1993	susanbrowng@email.com

PID.1	Email.2	State.1	State.2	Phone.1	Phone.2	Consent_Date.1	Consent_Date.2
3	susanbrowng@email.com	TN	TN	630-512-5824	630-512-5824	6/2/2017	9/14/2017
7	susanbrowng@email.com	TN	TN	630-512-5824	630-512-5825	9/14/2017	10/24/17
3	susanbrowng@email.com	TN	TN	630-512-5824	630-512-5825	6/2/2017	10/24/17
4	mjones@email.com	CA	CA	258-652-4875	258-652-4875	6/29/2017	10/20/2017
1	lsmith12@email.com	MA	FL	453-245-0712	828-304-4350	5/3/2017	9/4/2017
8	lsmith22@email.com	CA	MA	258-652-4875	453-245-0712	10/20/2017	5/3/2017
3	mjones@email.com	TN	CA	630-512-5824	258-652-4875	6/2/2017	6/29/2017
3	mjones@email.com	TN	CA	630-512-5824	258-652-4875	6/2/2017	10/20/2017
3	lsmith22@email.com	TN	MA	630-512-5824	453-245-0712	6/2/2017	5/3/2017
10	mjones@email.com	TN	CA	630-512-5825	258-652-4875	10/24/17	6/29/2017
3	lsmith12@email.com	TN	FL	630-512-5824	828-304-4350	6/2/2017	9/4/2017

Matched Set 4

PID.1	PID.2	ds	First_Name.1	First_Name.2	Last_Name.1	Last_Name.2	DOB.1	DOB.2	Email.1
3	7	6	susan	sue	brown	brown	8/29/1972	8/29/1972	susanbrowng@email.com
7	10	6	sue	sue	brown	brown	8/29/1972	8/29/1972	susanbrowng@email.com
3	10	9	susan	sue	brown	brown	8/29/1972	8/29/1972	susanbrowng@email.com

PID.1	Email.2	State.1	State.2	Phone.1	Phone.2	Consent_Date.1	Consent_Date.2
3	susanbrowng@email.com	TN	TN	630-512-5824	630-512-5824	6/2/2017	9/14/2017
7	susanbrowng@email.com	TN	TN	630-512-5824	630-512-5825	9/14/2017	10/24/17
3	susanbrowng@email.com	TN	TN	630-512-5824	630-512-5825	6/2/2017	10/24/17

Combine all matched sets into a single unique data frame

```
mm <- do.call("rbind", list(m1, m2, m3, m4))
mm <- unique(mm) %>% arrange(ds)
```

PID.1	PID.2	ds	First_Name.1	First_Name.2	Last_Name.1	Last_Name.2	DOB.1	DOB.2	Email.1
2	9	6	jennifer	jennifer	williams	william	8/18/1965	8/18/1966	jenniferwilliams@email.com
3	7	6	susan	sue	brown	brown	8/29/1972	8/29/1972	susanbrowng@email.com
7	10	6	sue	sue	brown	brown	8/29/1972	8/29/1972	susanbrowng@email.com
3	10	9	susan	sue	brown	brown	8/29/1972	8/29/1972	susanbrowng@email.com
4	8	10	michael	maria	jones	jones	10/24/1972	11/22/1972	mjones@email.com
1	6	22	linda	lidia	smith	smith	10/29/1964	11/3/1993	lsmith22@email.com
8	1	34	maria	linda	jones	smith	11/22/1972	10/29/1964	mjones@email.com
3	4	35	susan	michael	brown	jones	8/29/1972	10/24/1972	susanbrowng@email.com
3	8	36	susan	maria	brown	jones	8/29/1972	11/22/1972	susanbrowng@email.com
10	4	38	sue	michael	brown	jones	8/29/1972	10/24/1972	susanbrowng@email.com
3	1	38	susan	linda	brown	smith	8/29/1972	10/29/1964	susanbrowng@email.com
3	6	40	susan	lidia	brown	smith	8/29/1972	11/3/1993	susanbrowng@email.com

PID.1	Email.2	State.1	State.2	Phone.1	Phone.2	Consent_Date.1	Consent_Date.2
2	jenniferwilliams@email.com	PA	PA	462-946-0095	462-946-0095	5/10/2017	12/5/2017
3	susanbrowng@email.com	TN	TN	630-512-5824	630-512-5824	6/2/2017	9/14/2017
7	susanbrowng@email.com	TN	TN	630-512-5824	630-512-5825	9/14/2017	10/24/17
3	susanbrowng@email.com	TN	TN	630-512-5824	630-512-5825	6/2/2017	10/24/17
4	mjones@email.com	CA	CA	258-652-4875	258-652-4875	6/29/2017	10/20/2017
1	lsmith12@email.com	MA	FL	453-245-0712	828-304-4350	5/3/2017	9/4/2017
8	lsmith22@email.com	CA	MA	258-652-4875	453-245-0712	10/20/2017	5/3/2017
3	mjones@email.com	TN	CA	630-512-5824	258-652-4875	6/2/2017	6/29/2017
3	mjones@email.com	TN	CA	630-512-5824	258-652-4875	6/2/2017	10/20/2017
10	mjones@email.com	TN	CA	630-512-5825	258-652-4875	10/24/17	6/29/2017
3	lsmith22@email.com	TN	MA	630-512-5824	453-245-0712	6/2/2017	5/3/2017
3	lsmith12@email.com	TN	FL	630-512-5824	828-304-4350	6/2/2017	9/4/2017

Find optimal cut-point

Take random sample. In this example, we sample six observations.

```
set.seed(555)
s <- mm %>% sample_n(size = 6, replace = FALSE)
```

PID.1	PID.2	ds	First_Name.1	First_Name.2	Last_Name.1	Last_Name.2	DOB.1	DOB.2	Email.1
10	4	38	sue	michael	brown	jones	8/29/1972	10/24/1972	susanbrowng@email.com
2	9	6	jennifer	jennifer	williams	william	8/18/1965	8/18/1966	jenniferwilliams@email.com
3	4	35	susan	michael	brown	jones	8/29/1972	10/24/1972	susanbrowng@email.com
3	1	38	susan	linda	brown	smith	8/29/1972	10/29/1964	susanbrowng@email.com
4	8	10	michael	maria	jones	jones	10/24/1972	11/22/1972	mjones@email.com
3	10	9	susan	sue	brown	brown	8/29/1972	8/29/1972	susanbrowng@email.com

PID.1	Email.2	State.1	State.2	Phone.1	Phone.2	Consent_Date.1	Consent_Date.2
10	mjones@email.com	TN	CA	630-512-5825	258-652-4875	10/24/17	6/29/2017
2	jenniferwilliams@email.com	PA	PA	462-946-0095	462-946-0095	5/10/2017	12/5/2017
3	mjones@email.com	TN	CA	630-512-5824	258-652-4875	6/2/2017	6/29/2017

PID.1	Email.2	State.1	State.2	Phone.1	Phone.2	Consent_Date.1	Consent_Date.2
3	lsmith22@email.com	TN	MA	630-512-5824	453-245-0712	6/2/2017	5/3/2017
4	mjones@email.com	CA	CA	258-652-4875	258-652-4875	6/29/2017	10/20/2017
3	susanbrowng@email.com	TN	TN	630-512-5824	630-512-5825	6/2/2017	10/24/17

Manually annotate Data Set (truth)

```
s$truth <- c(0,1,0,0,0,1)
```

PID.1	PID.2	ds	First_Name.1	First_Name.2	Last_Name.1	Last_Name.2	DOB.1	DOB.2	Email.1
10	4	38	sue	michael	brown	jones	8/29/1972	10/24/1972	susanbrowng@email.com
2	9	6	jennifer	jennifer	williams	william	8/18/1965	8/18/1966	jenniferwilliams@email.com
3	4	35	susan	michael	brown	jones	8/29/1972	10/24/1972	susanbrowng@email.com
3	1	38	susan	linda	brown	smith	8/29/1972	10/29/1964	susanbrowng@email.com
4	8	10	michael	maria	jones	jones	10/24/1972	11/22/1972	mjones@email.com
3	10	9	susan	sue	brown	brown	8/29/1972	8/29/1972	susanbrowng@email.com

PID.1	Email.2	State.1	State.2	Phone.1	Phone.2	Consent_Date.1	Consent_Date.2	truth
10	mjones@email.com	TN	CA	630-512-5825	258-652-4875	10/24/17	6/29/2017	0
2	jenniferwilliams@email.com	PA	PA	462-946-0095	462-946-0095	5/10/2017	12/5/2017	1
3	mjones@email.com	TN	CA	630-512-5824	258-652-4875	6/2/2017	6/29/2017	0
3	lsmith22@email.com	TN	MA	630-512-5824	453-245-0712	6/2/2017	5/3/2017	0
4	mjones@email.com	CA	CA	258-652-4875	258-652-4875	6/29/2017	10/20/2017	0
3	susanbrowng@email.com	TN	TN	630-512-5824	630-512-5825	6/2/2017	10/24/17	1

Perform bootstrapping to find optimal cut-point In this example, we do 50 bootstrapped.

```
d <- data.frame(sim=as.numeric(), cutpoint=as.numeric()) # create empty data frame
for(i in seq(1, 50, 1))
{
  train_ind <- sample(seq_len(nrow(s)), size = 5, replace = F)
  train <- s[train_ind, ] #-- add observations to the training set
  test <- s[-train_ind, ] #-- add observations to the testing set

  train_cp <- optimal.cutpoints(X = "ds", status = "truth", tag.healthy = 1, methods = "Youden", data = train)
  train_cutoff <- as.numeric(train_cp$Youden$Global$optimal.cutoff$cutoff)
  test <- test[test$ds <= train_cutoff,]
  acc <- round(sum(test$truth) / length(test$PID.1), digits = 2)  #-- estimate accuracy on the test data set after filtering by the cutpoint found on the training data set
  temp <- data.frame(i,train_cutoff, acc)
  d <- rbind(d,temp) #-- add results of iteration to final dataset
}
```

Summary statistics of cut-point identified during bootstrapping

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
10	10	10	15	10	35

Summary statistics of the accuracy of each bootstrapped sample

Note: In extremely small samples (such as this one), it is possible to have no observations in the test data set after applying the cut-point identified in the training set. In those instances, it won't be possible to estimate accuracy.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
0	0	1	0.6551724	1	1	21

Note: At this point additional validation/manual verification is encouraged, but we are skipping it on this example.

Filter matched set to matches that are equal or less than the cut-point

PID.1	PID.2	ds	First_Name.1	First_Name.2	Last_Name.1	Last_Name.2	DOB.1	DOB.2	Email.1
2	9	6	jennifer	jennifer	williams	william	8/18/1965	8/18/1966	jenniferwilliams@email.com
3	7	6	susan	sue	brown	brown	8/29/1972	8/29/1972	susanbrowng@email.com
7	10	6	sue	sue	brown	brown	8/29/1972	8/29/1972	susanbrowng@email.com
3	10	9	susan	sue	brown	brown	8/29/1972	8/29/1972	susanbrowng@email.com
4	8	10	michael	maria	jones	jones	10/24/1972	11/22/1972	mjones@email.com

PID.1	Email.2	State.1	State.2	Phone.1	Phone.2	Consent_Date.1	Consent_Date.2
2	jenniferwilliams@email.com	PA	PA	462-946-0095	462-946-0095	5/10/2017	12/5/2017
3	susanbrowng@email.com	TN	TN	630-512-5824	630-512-5824	6/2/2017	9/14/2017
7	susanbrowng@email.com	TN	TN	630-512-5824	630-512-5825	9/14/2017	10/24/17
3	susanbrowng@email.com	TN	TN	630-512-5824	630-512-5825	6/2/2017	10/24/17
4	mjones@email.com	CA	CA	258-652-4875	258-652-4875	6/29/2017	10/20/2017

Even though the last row (Michael and Maria) fall within the mean cut-point of DS < = 15, just by looking at the data, we can tell that the PIDs belong to two different individuals. Instances like this should be manually excluded from the matched cohort.

Manual verification and correction

The last row of data is manually removed. Leaving the **final matched data set**

PID.1	PID.2	ds	First_Name.1	First_Name.2	Last_Name.1	Last_Name.2	DOB.1	DOB.2	Email.1
2	9	6	jennifer	jennifer	williams	william	8/18/1965	8/18/1966	jenniferwilliams@email.com
3	7	6	susan	sue	brown	brown	8/29/1972	8/29/1972	susanbrowng@email.com
7	10	6	sue	su	brown	brown	8/29/1972	8/29/1972	susanbrowng@email.com
3	10	9	susan	sue	brown	brown	8/29/1972	8/29/1972	susanbrowng@email.com

PID.1	Email.2	State.1	State.2	Phone.1	Phone.2	Consent_Date.1	Consent_Date.2
2	jenniferwilliams@email.com	PA	PA	462-946-0095	62-946-0095	5/10/2017	12/5/2017
3	susanbrowng@email.com	TN	TN	630-512-5824	630-512-5824	6/2/2017	9/14/2017
7	susanbrowng@email.com	TN	TN	630-512-5824	630-512-5825	9/14/2017	10/24/17
3	susanbrowng@email.com	TN	TN	630-512-5824	630-512-5825	6/2/2017	10/24/17

Disjoint sets implementation

The following code is an implementation of the mathematical algorithm for disjoint sets and it was taken from: https://github.com/malllabiisc/cesi/blob/master/src/unionFind.py

Note: This section is written in Python and not R.

```
class DisjointSet(object):

    def __init__(self):
        self.leader = {} # maps a member to the group's leader
        self.group = {} # maps a group leader to the group (which is a set)

    def add(self, a, b):
        leadera = self.leader.get(a)
        leaderb = self.leader.get(b)
        if leadera is not None:
            if leaderb is not None:
                if leadera == leaderb: return # nothing to do
                groupa = self.group[leadera]
                groupb = self.group[leaderb]
                if len(groupa) < len(groupb):
                    a, leadera, groupa, b, leaderb, groupb = b, leaderb, groupb, a, leadera, groupa
                groupa |= groupb
                del self.group[leaderb]
                for k in groupb:
                    self.leader[k] = leadera
            else:
                self.group[leadera].add(b)
                self.leader[b] = leadera
        else:
            if leaderb is not None:
                self.group[leaderb].add(a)
                self.leader[a] = leaderb
            else:
                self.leader[a] = self.leader[b] = a
                self.group[a] = set([a, b])
```

Disjoint sets execution on final matched set

Note: This section is written in Python and not R.

```
import pandas as pd
matches = pd.DataFrame(r.matched_set) #-- r. is needed to access an R object

mylist = [] #-- initialize list
for item in matches.values:
    mylist.append(list(item))  #-- add all the values in the final matched set to a list

ds = DisjointSet() #-- Disjoint set object
final_set = set() #-- Disjoint data set

for element in mylist:
    ds.add(element[0], element[1])

for e in ds.group:
    final_set.add(frozenset(ds.group[e])) #-- add all the unions found to output set

df = pd.DataFrame(list(final_set)) #-- convert disjoint set to a data frame
col_names = []
for col in range(0,len(df.columns)): #-- rename columns
    col_names.append("pid."+str(col+1))
df.columns = col_names
```

Disjoint Data Set

A truly unique ID can now be added to identify each individual, prevent duplication and link orphaned data together.

unique_id	pid.1	pid.2	pid.3
1	3	7	10
2	9	2	NULL

Final Disjoint Data Set

To create the final disjoint data set, we also need to take into account those PIDs not represented in the disjoint data set because these were found not to be duplicates. Unique PIDs and PIDs contained in the disjoint data set are then joined into a final disjoint data set where a truly unique identifier is assigned to each individual.

```r
#-- convert disjoint data set to long format
dds_long <- dds %>%
  gather("name", "PID", -unique_id) %>%
  select("unique_id","PID") %>%
  arrange("unique_id") %>%
  filter(PID != "NULL") %>%
  mutate(PID = sapply(PID, toString)) %>%
  as.data.frame()

#-- get PIDs in the original data not contained in the disjoint dataset and assign sequencial new IDs
not_in_dds <- data %>% filter(!PID %in% dds_long$PID) %>% add_column(unique_id = NA) %>%select(unique_id, PID) %>% as.data.frame()
not_in_dds <- not_in_dds %>% mutate(unique_id = max(dds_long$unique_id)+1:n()) %>% mutate(PID = sapply(PID, toString))

#-- Put both datasets together and merge with the original data
dds_long <- rbind(dds_long, not_in_dds) %>% arrange("unique_id")
final_dds <- merge(dds_long, data, by=c("PID"),all=T)
```

In this example, we found two duplicate individuals and established that we actually have seven unique individuals in the data.

unique_id	PID	DID	First_Name	Last_Name	DOB	Email	State	Phone	Consent_Date
1	10	3	sue	brown	8/29/1972	susanbrowng@email.com	TN	630-512-5825	10/24/17
1	3	3	susan	brown	8/29/1972	susanbrowng@email.com	TN	630-512-5824	6/2/2017
1	7	3	sue	brown	8/29/1972	susanbrowng@email.com	TN	630-512-5824	9/14/2017
2	2	2	jennifer	williams	8/18/1965	jenniferwilliams@email.com	PA	462-946-0095	5/10/2017
2	9	2	jennifer	william	8/18/1966	jenniferwilliams@email.com	PA	462-946-0095	12/5/2017
3	1	1	linda	smith	10/29/1964	lsmith22@email.com	MA	453-245-0712	5/3/2017
4	4	4	michael	jones	10/24/1972	mjones@email.com	CA	258-652-4875	6/29/2017
5	5	5	james	davis	7/20/1988	jamesdavis44@email.com	TX	880-391-9208	7/19/2017
6	6	6	lidia	smith	11/3/1993	lsmith12@email.com	FL	828-304-4350	9/4/2017
7	8	4	maria	jones	11/22/1972	mjones@email.com	CA	258-652-4875	10/20/2017