Preparation of a real temporal voter data set for record linkage and duplicate detection research

Peter Christen

Research School of Computer Science, College of Engineering and Computer Science
The Australian National University, Canberra ACT 0200, Australia
peter.christen@anu.edu.au

Last update: 29 June 2014

Abstract. This report describes the process involved in accessing, processing, and merging a set of files containing voter registration information from the US state of North Carolina (NC). We have downloaded these files every two months since October 2011, and combined them into one temporal data set, which we name the *NC Voter Registration* (NCVR) data set.
Each individual voter in this longitudinal NCVR data set is represented by one or more records, and each voter is given a unique identifier number. For each voter only records are included into the final longitudinal data set if these records are not exact duplicates of each other, i.e. records for a voter are only added to the temporal data set if some attribute values have changed in a record compared to the values in a previous record (downloaded earlier) of the same voter. At the time of writing, the NCVR data set has been downloaded and processed seventeen times (bi-monthly since October 2011), resulting in a compound data set that contains nearly 8.3 million records of over 8 million individual voters, with around 98% of voters represented by a single record, around 145,000 by two records, and around 3,500 voters are represented by three to six records.
As a result of the conducted processing and merging, this NCVR data set contains the personal information of a large number of individuals, and their changes in names, addresses and other personal details over time, as well as instances where data entry errors and other smaller variations have been corrected. Because the identity of voters is available through a voter registration number (although not 100% correct), the NCVR data set is a highly valuable resource for research in areas such as record linkage and duplicate detection that crucially rely upon the availability of real personal information and truth data about which records represent each individual.

1 Background

Much research in recent times has been conducted in the areas of data mining and data matching (also known as record linkage, entity resolution, or duplicate detection) [1,2,3,4]. With a large portion of today's data being collected about or by people, having access to real-world data sets that contain the personal details of a large number of individuals, and the changes of these personal details over time, can be a valuable research resource. However, because personal information stored in government and business databases is commonly protected and cannot be published, it is difficult for researchers to get access to large databases that contain personal information.

An exception to the publication of personal data are voter registration data-bases, which in several countries are publicly available (however, most often not in electronic format or not online). In some US states, voter databases can be purchased either on a record-by-record basis, or as a full database, while in other states and countries such data can only by viewed in person in a voter registration or electoral office. One US state which has made its complete voter registration database freely and publicly available via an FTP server is North Carolina (NC). The website of the NC State Board of Elections (NCSBE, `http://www.ncsbe.gov/`) provides a regular update of this database[1].

The complete list of all registered NC voters is available as a collection of 100 text files (in a tabulator separated values format) as well s more recently as a single state-wide file, each containing nearly 70 different attributes (as listed in the Appendix). These include, for each record, information about the status of a voter's

[1] See: `ftp://alt.ncsbe.gov/data/`

Table 1. Basic download statistics for concatenated NCVR files (after removal of records with incorrect dates).

Download date	Size of file	Num of records	Num of distinct VRNs	Num of VRNs multiple records	Num of incorrect registration dates
4 Oct 2011	635 MB	6,233,685	2,802,362	700,188	19,312
3 Dec 2011	711 MB	6,981,777	3,013,266	762,477	23,213
4 Feb 2012	710 MB	6,974,907	3,020,349	726,251	22,901
1 Apr 2012	718 MB	7,054,742	3,059,669	773,416	22,881
2 Jun 2012	722 MB	7,090,380	3,078,797	778,072	22,871
4 Aug 2012	727 MB	7,134,354	3,100,312	781,221	22,827
1 Oct 2012	745 MB	7,310,235	3,203,187	805,906	22,766
3 Dec 2012	766 MB	7,524,477	3,307,441	834,754	22,720
2 Feb 2013	739 MB	7,251,819	3,235,939	815,213	21,350
3 Apr 2013	741 MB	7,268,065	3,244,340	816,353	20,604
2 Jun 2013	743 MB	7,291,727	3,255,845	819,006	19,428
1 Aug 2013	747 MB	7,325,037	3,272,587	822,865	18,890
1 Oct 2013	750 MB	7,358,267	3,287,470	826,857	18,262
3 Dec 2013	753 MB	7,388,105	3,304,359	830,252	17,942
2 Feb 2014	754 MB	7,391,222	3,304,268	830,444	31,889
5 Apr 2014	642 MB	6,293,509	2,774,253	745,734	60,736
5 Jun 2014	642 MB	7,453,885	3,299,779	829,086	109,572

registration, the voter's name and address details, their age, gender, drivers license and telephone numbers, as well as their race, ethnicity and registered party affiliations. Not all of this information is available in all records, as is shown in Table **??** below.

Only some of these attributes are of value for research in the areas of record linkage and duplicate detection, where the main objective is to use personal identifiers, such as names and address details, to identify and match records that refer to the same entity (in the NCVR data set these entities are individual voters).

In the following section we describe in detail the characteristics of the attributes selected for the temporal NC voter data set we generated, while in Section 3 we describe how we cleaned and pre-processed, and then combined, the individual data sets downloaded on a bi-monthly basis since October 2011.

2 File download and basic characteristics

The individual text files that describe the current details of all voters in NC have been downloaded from the URL given in Footnote 1 on a regular basis since October 2011. Table 1 shows the exact dates when the NCVR files were downloaded, their sizes, as well as some basic statistics of these files. As can be seen, a significant number of voter registration numbers (VRN) occurred more than once at any download date.

Table 2 shows the frequency distributions of how many times a distinct VRN occurred. VRNs that occurred several times are processed as will be described in the following section.

We decided – in a somewhat arbitrary way – to set all registration dates with a value before the 1 January 1930 (over 80 years ago) and those with a value after the current date as *incorrect* dates. We argue that voters with registration dates before 1930 would have to be over 100 years old now, so removing voters with older registration dates would only affect a small number of voters. We also set as incorrect dates those with a value in an invalid date format. All records with incorrect registration dates are deleted from the downloaded files prior to further processing. As can be seen from Table 1, the number of incorrect dates increases steadily over time.

The following Tables 3, 4, 5, and 6 show the basic data characteristics of these individual files as the number of unique values in an attribute, as well as the percentage of records that have a missing (empty) value in an attribute. Values in the age attribute are integer values. In the gender attribute we have three values: 'f', 'm', and 'u' (for an unknown gender). The values in the birth_place attribute are two-letter US state abbreviations. The file downloaded in April 2014 was severely corrupted, resulting in many more unique values in the majority of attributes.

Table 2. Distribution of number of times a distinct VRN occurred in a downloaded file.

Download	Frequency of VRN occurrence						Maximum
date	1	2	3	4	5–9	10+	frequency
4 Oct 2011	2,102,174	295,859	95,900	72,773	117,671	117,985	53
3 Dec 2011	2,250,789	328,241	93,714	75,517	131,906	133,099	57
4 Feb 2012	2,258,098	327,010	94,121	76,213	132,244	132,663	57
1 Apr 2012	2,286,253	332,179	95,534	77,167	134,622	133,914	56
2 Jun 2012	2,300,725	333,701	96,552	78,224	135,281	134,314	56
4 Aug 2012	2,319,091	332,972	96,661	79,449	137,043	135,096	56
1 Oct 2012	2,397,281	345,218	100,549	82,644	140,963	136,532	56
3 Dec 2012	2,472,687	357,928	102,863	85,633	149,133	139,197	56
2 Feb 2013	2,420,726	346,201	104,051	87,251	144,904	132,806	55
3 Apr 2013	2,427,987	345,542	104,221	87,898	145,632	133,060	55
2 Jun 2013	2,436,839	346,192	104,398	88,442	146,570	133,404	55
1 Aug 2013	2,449,722	347,350	104,470	89,415	147,780	133,850	55
1 Oct 2013	2,460,613	348,434	104,781	90,421	148,849	134,372	55
3 Dec 2013	2,474,107	349,054	105,696	91,150	149,662	134,690	54
2 Feb 2014	2,473,824	348,932	106,155	91,499	149,496	134,362	54
5 Apr 2014	2,028,519	357,139	112,481	64,275	92,265	119,574	54
5 Jun 2014	2,470,693	348,002	106,712	91,554	148,788	134,030	54

3 Data processing

In this section we described the three main steps taken to processing the raw NCVR files (the ones summarised in Table 1) and merge them into a single temporal data set. Figures 1 to 3 outline the main sub-steps involved in each of the three main processing steps. In the first step (*Process individual files*), each file is processed separately, and exact duplicates records are removed. In the second step (*Remove duplicates across files*), the individual files are ordered according to their download dates, and records are compared between individual files. Records in later files that are identified as exact duplicates of a record in an earlier file are removed.

In the third and final step (*Merge records across files*), for each voter a list of one or more records is generated based on a series of similarity tests on selected attribute combinations. Finally, the resulting single data set is written into a CSV file. Details of these processing steps are given in the following sub-sections. The programs used for cleaning and processing the NCVR data set are all written in the Python[2] programming language.

3.1 Step 1: Individual file pre-processing

In this step, each NCVR file is processed individually, with the aim to identify exact duplicates records, both those that have the same VRN but also those that have different VRNs. Figure 1 outlines the steps conducted on each file.

We designate exact duplicate records as those that have the same values in selected attribute check combinations (exact checks). Table 7 shows the different combinations of attributes we consider. With the exception of voter_reg_num, name_prefix, birth_place, register_date, and the four code attributes (status, reason, race, and ethnicity), we consider all attributes that contain meaningful information about individual voters that can help distinguish records of one voter from those of another.

For each column of Table 7, the values in the attributes shown with a $\otimes$ are concatenated into a one string (i.e. one string will represent one record). Records that have the same string are designated as exact duplicates. As described in Figure 1, in cases where exact duplicate records with the same VRN occur we only keep one of them (step 1.2), and exact duplicates with different VRNs we only keep the record with the smallest VRN (step 1.4).

The age attribute is handled in a special case, as even within a single NCVR file there can be duplicate records of the same voter with the age value differing by 1 year. We therefore conduct all checks E1 to E7 by not considering the age value of records when the check string of a record is generated, but we rather

[2] `http://www.python.org`

Table 3. Number of unique values in attributes in concatenated NCVR files (part 1).

Attribute name	Oct 2011	Dec 2011	Feb 2012	Apr 2012	Jun 2012	Aug 2012	Oct 2012	Dec 2012	Feb 2013	Apr 2013	Jun 2013
voter_reg_num	2,802,692	3,013,625	3,020,704	3,060,028	3,079,158	3,100,669	3,203,540	3,307,795	3,236,286	3,244,693	3,256,199
status_code	4	5	5	5	5	5	5	5	5	5	5
reason_code	23	31	31	31	31	31	31	31	30	30	30
name_prefix	17	20	0	0	0	0	0	0	0	0	0
first_name	193,552	203,296	204,576	208,166	209,429	211,032	219,053	226,733	223,520	224,380	225,316
middle_name	274,385	291,708	293,540	298,655	300,387	302,708	312,181	320,259	315,113	316,433	317,875
last_name	281,704	294,823	295,966	298,896	300,431	302,165	311,765	320,110	313,780	314,806	316,125
name_suffix	14	15	15	15	15	16	16	16	16	16	16
age	105	107	110	111	113	113	116	125	108	106	104
gender	3	3	3	3	3	3	3	3	3	3	3
race_code	7	7	7	7	7	7	7	7	7	7	7
ethnicity_code	3	3	3	3	3	3	3	3	3	3	3
res_street_address	3,258,731	3,434,732	3,435,917	3,451,715	3,460,686	3,473,126	3,522,873	3,581,569	3,518,417	3,523,693	3,528,860
city	784	792	791	790	790	790	789	789	784	785	785
state	4	4	4	4	4	4	4	4	4	4	4
zip_code	892	906	904	902	904	899	898	897	888	886	886
full_phone_num	1,926,803	2,080,526	2,088,744	2,120,672	2,131,362	2,151,179	2,219,482	2,279,906	2,227,706	2,241,393	2,253,747
birth_place	58	58	58	58	60	60	60	60	60	60	60
register_date	20,053	20,593	20,634	20,697	20,748	20,809	20,865	20,924	20,822	20,830	20,839

Table 4. Number of unique values in attributes in concatenated NCVR files (part 2).

Attribute name	Aug 2013	Oct 2013	Dec 2013	Feb 2014	Apr 2014	Jun 2014
voter_reg_num	3,272,942	3,287,823	3,304,714	3,311,218	2,798,630	3,341,378
status_code	5	5	5	5	78	5
reason_code	30	30	30	30	107	30
name_prefix	0	0	0	0	81	0
first_name	226,789	227,954	229,077	229,746	202,397	232,439
middle_name	320,163	322,125	323,924	325,144	285,967	329,215
last_name	317,822	319,871	321,841	322,905	283,850	325,972
name_suffix	16	16	16	16	117	18
age	104	105	105	105	213	107
gender	3	3	3	3	119	3
race_code	7	7	7	8	124	7
ethnicity_code	3	3	3	3	109	3
res_street_address	3,536,001	3,544,675	3,551,969	3,553,853	3,108,168	3,567,915
city	784	783	782	783	874	782
state	4	4	4	4	105	4
zip_code	884	883	881	881	977	881
full_phone_num	2,269,527	2,288,820	2,302,652	2,314,399	2,015,354	2,350,105
birth_place	60	60	60	60	171	58
register_date	21,025	21,094	21,153	21,185	20,950	21,200

Table 5. Percentage of records with missing values in attributes in concatenated NCVR files (part 1).

Attribute name	Oct 2011	Dec 2011	Feb 2012	Apr 2012	Jun 2012	Aug 2012	Oct 2012	Dec 2012	Feb 2013	Apr 2013	Jun 2013
voter_reg_num	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
status_code	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
reason_code	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
name_prefix	99.98%	99.98%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
first_name	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
middle_name	6.47%	6.50%	6.51%	6.49%	6.49%	6.48%	6.54%	6.75%	6.75%	6.73%	6.72%
last_name	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
name_suffix	94.38%	94.36%	94.36%	94.36%	94.36%	94.36%	94.35%	94.33%	94.33%	94.32%	94.32%
age	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
gender	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
race_code	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
ethnicity_code	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
res_street_address	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
city	0.01%	0.01%	0.01%	0.01%	0.01%	0.01%	0.01%	0.01%	0.01%	0.01%	0.01%
state	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
zip_code	0.01%	0.01%	0.01%	0.01%	0.01%	0.01%	0.01%	0.01%	0.01%	0.01%	0.01%
full_phone_num	59.24%	59.83%	59.73%	59.64%	59.72%	59.69%	59.69%	60.05%	59.85%	59.76%	59.70%
birth_place	14.70%	15.19%	15.08%	14.96%	14.97%	14.99%	15.58%	16.33%	16.10%	16.01%	15.94%
register_date	0.31%	0.33%	0.33%	0.32%	0.32%	0.32%	0.31%	0.30%	0.45%	0.91%	1.3%

Table 6. Percentage of records with missing values in attributes in concatenated NCVR files (part 2).

Attribute name	Aug 2013	Oct 2013	Dec 2013	Feb 2014	Apr 2014	Jun 2014
voter_reg_num	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
status_code	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
reason_code	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
name_prefix	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
first_name	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
middle_name	6.71%	6.71%	6.70%	6.69%	6.54%	6.68%
last_name	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
name_suffix	94.33%	94.33%	94.33%	94.34%	94.36%	94.34%
age	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
gender	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
race_code	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
ethnicity_code	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
res_street_address	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
city	0.01%	0.01%	0.01%	0.01%	0.01%	0.01%
state	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
zip_code	0.01%	0.01%	0.01%	0.01%	0.01%	0.01%
full_phone_num	59.65%	59.56%	59.54%	59.42%	58.47%	59.24%
birth_place	15.87%	15.78%	15.69%	16.60%	15.05%	15.46%
register_date	0.26%	0.25%	0.24%	0.43%	0.97%	1.47%

Step 1: **Process individual files**

For each individual NCVR downloaded file shown in Table 1 do:

- Step 1.1: For each distinct VRN in a file, build a list of all records with this VRN.
- Step 1.2: For all VRNs that have more than one record:
  - Check if records are exact duplicates of each other according to one of the attribute check combinations from Table 7.
  - For all exact duplicate records only keep one of them.
- Step 1.3: For all VRNs that still have more than one record:
  - Assign each of these record a new unique VRN, and remove the original VRN and its records from the list of all VRNs.
- Step 1.4: For each attribute check combination from Table 7:
  - Find all records that have the same value for the selected attribute combination.
  - If several records have the same attribute check combination value, remove all except the one with the smallest VRN.
- Step 1.5: Write the remaining list of VRNs (each with one record) into a new file named after download date.

Fig. 1. Step involved in processing each file individually.

Table 7. Attribute check combinations used to identify exact duplicate records within and across files ('E' stands for 'exact' check). Each column refers to one combination of attributes used (the ones with a ⊗). The age attribute (indicated by ⊙) is handled in a special way, because age values are changing over time. Details are described in Sections 3.1 and 3.2.

Attribute name	Attribute combinations used																
	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14	E15	E16	E17
first_name	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
middle_name	⊗	⊗	⊗	⊗	⊗			⊗	⊗	⊗			⊗				
last_name	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
name_suffix	⊗	⊗	⊗		⊗	⊗	⊗	⊗	⊗	⊗			⊗	⊗	⊗		⊗
age	⊙	⊙	⊙	⊙	⊙	⊙	⊙			⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
gender	⊗		⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗		⊗	⊗	⊗	⊗	⊗
race_code													⊗				
ethnicity_code													⊗				
res_street_address	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗		⊗	⊗		⊗	⊗	⊗	
city	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗		⊗		⊗	⊗
state	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗		⊗	⊗	⊗	⊗
zip_code	⊗	⊗	⊗	⊗		⊗	⊗	⊗	⊗	⊗	⊗	⊗			⊗		⊗
full_phone_num	⊗	⊗		⊗	⊗	⊗		⊗								⊗	⊗
birth_place													⊗				
register_date													⊗				

generate the check strings without the age value and then calculate the numerical age difference for records with the same attribute check combination value. Those records that have an age difference of maximum 1 year are designated as duplicates and processed in the same way as exact duplicates (as outlined in Figure 1). A special case is records where the age values is 111 (which seems to be an indicator of a missing age value, given its high overall frequency in the NCVR files). If one of the two records that are compared for exact duplicate status has an age value of 111 and the other record has an age of 50 or above we also designate the records as duplicates and remove one of the two records.

At the end of the first processing step, the data for each individual NCVR file consist of a list of VRNs with one record each, such that none of these records is an exact duplicate of any other record in the same file according to the attribute check combinations shown in Table 7.

Table 8. Number of distinct VRNs at the end of processing step 1, and the number of duplicate records (according to the attribute check combinations shown in Table 7) that have been deleted.

Download date	Number of unique VRNs	Number of deleted records
4 Oct 2011	6,194,106	20,459
3 Dec 2011	6,914,899	43,665
4 Feb 2012	6,912,150	39,856
1 Apr 2012	6,991,575	40,286
2 Jun 2012	7,026,716	40,793
4 Aug 2012	7,070,119	41,408
1 Oct 2012	7,243,954	43,515
3 Dec 2012	7,453,027	48,730
2 Feb 2013	7,190,038	40,431
3 Apr 2013	7,207,343	40,118
2 Jun 2013	7,232,033	40,266
1 Aug 2013	7,265,751	40,396
1 Oct 2013	7,299,493	40,512
3 Dec 2013	7,329,429	40,734
2 Feb 2014	7,319,503	39,830
5 Apr 2014	6,199,697	33,076
5 Jun 2014	7,304,809	39,504

3.2 Step 2: Removing duplicates across files

Based on the individual lists of VRNs and records generated in the first processing step, in this second step we aim to remove duplicate records across files. These are records of voters where none of their details have changed (according to the attribute check combinations shown in Table 7. Figure 2 outlines the individual steps conducted in this process.

The basic idea is that records in each individual file (downloaded at a certain point in time) are compared with records in files that have been downloaded earlier (step 2.1), and the same checks for exact duplicates as in step 1 are carried out (step 2.2). We again use the attribute check criteria shown in Table 7, first on records that have the same VRN across the two files (step 2.3), and then for records across the two files that have different VRNs but the same values in an attribute check combination (step 2.4).

Because we now compare files over longer period of time, when we do include values from the age attribute (E1 to E7) for checking we allow for a (currently) maximum age difference of 4 years as we do have NCVR files downloaded from 2011 to 2013. We again handle records with an age value of 111 in a special as was discussed above.

Table 9 shows the number of duplicate records deleted (and how many of these duplicates had the same or a different VRN) and the number of different records that are kept for the merging of the individual files in the third processing step.

It is interesting to note that a large number of exact duplicates have a different VRN. It can also be seen that the larger the time difference between the earliest and a following file, the more records have different attributes values. This is expected, because over time more people will change their address, telephone number, or personal name.

At the end of the second processing step, the data for the earliest NCVR file (downloaded in October 2011) consists of a list of VRNs with one record each (the base list), such that none of these records is a duplicate of any other record according to the attribute check combinations shown in Table 7). For all NCVR files downloaded later on, we also have a list of VRNs with one record each, but these lists only contain VRNs where their records do not have a duplicate record (according to the attribute check combinations from Table 7) with any record in an earlier NCVR file In the third processing step these lists are merged into one final longitudinal data set.

3.3 Step 3: Merging records across files

The objective of this final processing step is to merge records from the individual NCVR files into one final longitudinal data set, by appending records that correspond to the same voter to the correct VRN, such that

Step 2: **Remove duplicates across files**

For each individual NCVR downloaded file shown in Table 1 do:
- Step 2.1: For each earlier NCVR file (according to download month):
  - Step 2.2: For each distinct VRN that occurs both in the current and the earlier (according to download month) file:
    - Check if the records for this VRNs are exact duplicates between the current and earlier file according to one of the attribute check combinations from Table 7.
    - Remove any record with an exact duplicate in the earlier file from the list of the current file.
  - Step 2.3: For each attribute check combination from Table 7:
    - Find all records in the current and earlier file that have the same value in the selected attribute combination.
    - If several records have the same attribute check combination value, remove the records in the list of the current file.
  - Step 2.4: Write the remaining list of VRNs (each with one record) of the current file into a new file named after download date.

Fig. 2. Step involved in removing duplicates across files.

Table 9. Number of VRNs in the NCVR files other than the earliest (October 2011) that have a duplicate in an earlier NCVR file according to the attribute check combinations shown in Table 7 (but with the age attribute comparison relaxed as described in Section 3.2). Also shown is how many of these duplicates had the same VRN or a different VRN. Only records with different values (last column) are kept for further processing in step 3.

Download date	Number of exact duplicates		Number of records with
	Same VRN	Different VRN	different values
3 Dec 2011	2,023,127	4,148,383	743,389
4 Feb 2012	2,207,751	4,642,421	61,978
1 Apr 2012	2,211,821	4,663,425	116,329
2 Jun 2012	2,243,524	4,713,096	70,096
4 Aug 2012	2,249,682	4,738,917	81,520
1 Oct 2012	2,219,444	4,750,914	273,596
3 Dec 2012	2,266,237	4,832,911	353,879
2 Feb 2013	2,328,955	4,803,291	57,792
3 Apr 2013	2,341,632	4,820,508	45,203
2 Jun 2013	2,350,937	4,839,768	41,328
1 Aug 2013	2,357,901	4,857,234	50,616
1 Oct 2013	2,367,001	4,875,323	57,169
3 Dec 2013	2,376,034	4,897,732	55,663
2 Feb 2014	2,388,213	4,902,075	29,215
5 Apr 2014	1,892,901	4,291,300	15,496
5 Jun 2014	2,389,069	4,899,495	16,245

each voter is represented by one or more records. Because exact duplicates records have been removed in the first two processing steps, only records that have different values in some attributes are assigned to a voter. Figure 3 outlines the step involved in this merging process.

Similar to the exact duplicate checks based on the attribute combination checks shown in Table 7, in this step we use a set of attribute checks to identify candidate records that are likely refer to the same voter. As Table 10 shows, these attribute combinations are more relaxed than the checks for exact duplicates and allow for more attributes to have different values.

Records that have the same value in the selected attributes for one of these check combinations will be further compared with the record(s) in the base list that have the same check value. These comparisons are based on similarity comparisons of attribute values rather than exact comparisons only, as shown in Figure 4. For the attributes that contain strings (all except age), we use a bi-gram based approximate string

Table 10. Attribute check combinations used to identify which records to add/merge to the record list of a VRN ('M' stands for 'merge' check). Each column refers to one combination of attributes used (the ones with a ⊗). The age attribute (indicated by ⊙) is handled in a special way, because age values are changing over time. Details are described in Sections 3.3.

Attribute name	Merge attribute combinations used																			
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16	M17	M18	M19	M20
first_name	⊗	⊗	⊗	⊗		⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
middle_name	⊗	⊗	⊗	⊗	⊗			⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗			
last_name	⊗			⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
name_suffix	⊗	⊗	⊗		⊗		⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
age		⊙	⊙	⊙		⊙		⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
gender	⊗	⊗	⊗					⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
race_code															⊗			⊗		
ethnicity_code																⊗			⊗	
res_street_address	⊗	⊗	⊗	⊗	⊗	⊗	⊗													
city	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗		⊗		⊗						
state	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗						
zip_code	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗		⊗		⊗							
full_phone_num		⊗						⊗			⊗	⊗								
birth_place															⊗	⊗	⊗	⊗	⊗	⊗
register_date																	⊗			⊗

Step 3: **Merge records across files**

- Step 3.1: Build the base list using the VRNs and their records from the earliest NCVR file.
- Step 3.2: For each following NCVR file (according to download month):
  - Step 3.3: For each distinct VRN that occurs both in the base list and the current file:
    - Check if the records for this VRNs fulfil one of the merge criteria (Table 10 plus similarity checks), if they do append the record from the current NCVR file to the record current NCVR file to the record list of the VRN in the base list.
  - Step 3.4: For each attribute check combination from Table 10:
    - Find records in the current NCVR file that have the same value in the selected attribute combination as a record in the base list.
    - If similarity checks find a matching pattern, append record from current file to the matching list of records in the base list.
  - Step 3.5: For all records from the current NCVR file that have not been matched to a VRN in the base list:
    - Assign each of these record a new unique VRN, and add them into the base list.
  - Step 3.6: Write the base list (VRNs and their assigned list of records) into the final longitudinal NCVR data set.

Fig. 3. Step involved in merging records across files.

comparisons function, and for age values we calculate their difference as percentage over the larger of two age values [1]. All similarity values are normalised in $[0.0, \ldots, 1.0]$, and two attribute values that have a similarity above 0.6 are designated to be the same, and different otherwise.

Nearly twenty different such similarity checks are applied on any pair of records that have the same attribute combination according to one of the checks from Table 10, and if any of these similarity checks classify a record from a later NCVR file to be matching with a record in the base list (i.e. an earlier file), then the record is added to the corresponding list in the base list. Similar as is done in processing steps 1 and 2, as detailed in Figure 3, we first check records for similarities that have the same VRN (step 3.3) and then records with different VRNs (step 3.4). All records that do not fit any of the merging check criteria from Table 10 are assumed not to refer to an already known voter in the base list, and are thus given a unique new VRN and then added to the base list (step 3.5).

The final step is to write the base list, made of VRNs and their lists of records, into the final longitudinal NCVR data set. This data set can either be sorted according to the VRNs, the registration and download dates, or be randomly shuffled. The values in the voter_reg_num attribute (many of these created during the three processing steps) are not written into the final longitudinal NCVR data set, rather each voter (and its records) is given a unique voter identifier voter_id in the form of a 8-digit integer number (starting from 0).

Table 11 shows the final distribution of the number of voters with a certain number of records attached. As can be seen, the vast majority of voters (over 98%) is represented by a single record only). Figures 5 shows the distribution of registration dates since 1930.

4 Conclusions

This paper described the steps involved in obtaining and preparing a large data set containing the personal details of over 8 million voters in the US state of North Carolina (NC). By downloading this data set every two months since October 2011, we were able to generate a comprehensive combined longitudinal data set that contains information about how people change their personal details, such as their name and addresses,

Similarity check criteria: A record in a NCVR file is added to the list of records of a VRN in the base list if any of the following criteria is true:

1. Same first_name, middle_name, last_name, age, gender, and city.
2. Same first_name, last_name, age, gender, and full_phone_num.
3. Same first_name, middle_name, last_name, age, and res_street_address.
4. Same first_name, middle_name, age, city, and res_street_address.
5. Same first_name, middle_name, last_name, age, and gender; and name_suffix must be empty in both records (i.e. no different middle name).
6. Same first_name, last_name, name_suffix, age, gender; and middle_name must be empty in both records (i.e. no different name suffix).
7. Same first_name, last_name, age, gender, and city; and middle_name and name_suffix must both be empty in both records (i.e. no different middle name or name suffix).
8. Same first_name, last_name, age, gender, and city; and middle_name has the same first letter (i.e. same initials) and name_suffix must be empty in both records (i.e. no different name suffix).
9. Same middle_name, last_name, age, gender, res_street_address, and city; and first_name has the same first letter (i.e. same initials) and name_suffix must be empty in both records (i.e. no different name suffix).
10. Same first_name, middle_name, age, gender, and city; and name_suffix must be empty in both records (i.e. no different name suffix).
11. Same first_name, middle_name, last_name, name_suffix, age, and gender.
12. Same last_name, age, gender. res_street_address, and city; and first_name and middle_name values are not empty, and sorted and concatenated into one string they are the same in both records.
13. Same first_name, last_name, age, res_street_address, and city; and middle_name and name_suffix must both be empty in both records (i.e. no different middle name or name suffix).
14. Same first_name, last_name, name_suffix, age, gender, res_street_address, and city; and at least one of middle_name must be empty (i.e. no conflicting middle names).
15. Same first_name, last_name, name_suffix, age, gender, res_street_address, and city; and both middle_name values are not empty and one must be a sub-string of the other (i.e. must be contained in the other).
16. Same first_name, last_name, age, res_street_address, city, and zip_code; and either at least one name_suffix value is empty or they are the same; and either at least one middle_name value is empty or they have the same first letter; and either the gender values are the same or at least one of them is 'u' (unknown).
17. Same first_name, age, res_street_address, city, and zip_code; and the gender value is 'f' (female); and middle_name and name_suffix must both be empty in both records (i.e. no different middle name or name suffix).
18. Same last_name, age, gender, res_street_address, and city; and the concatenated first_name and middle_name are the same in both records.
19. Same first_name, last_name, age, res_street_address, city, and zip_code; and both middle_name and name_suffix are not different (i.e. the same or at least one of the two values is empty).

Fig. 4. Similarity check criteria for merging records used in processing step 3.

as well as about data entry errors and variations. This data set can be a valuable resource for research in areas such as record linkage and duplicate detection.

Table 11. Final distribution of records per voter.

Number of records per voter	Number of voters with that many records
1	7,962,101
2	144,468
3	3,476
4	88
5	3
6	1

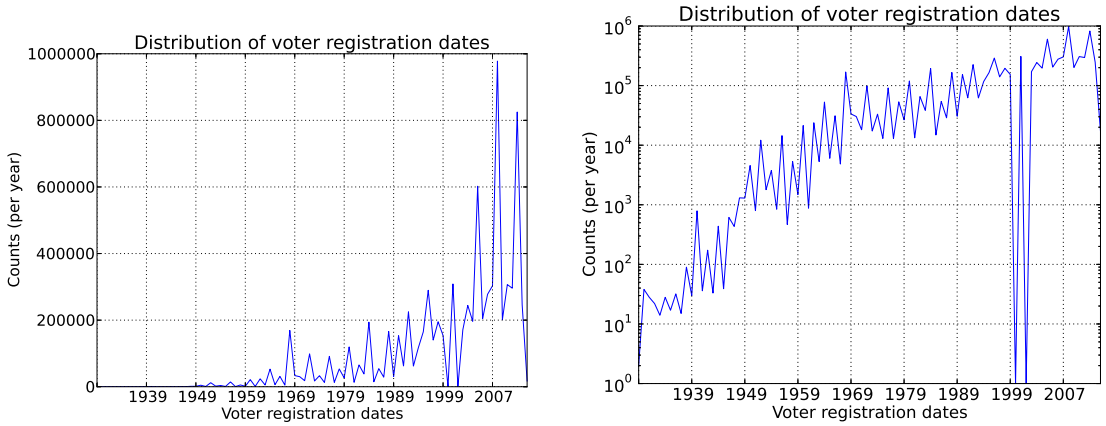

Fig. 5. Distribution of the voter registration dates, summed on a yearly basis and shown with a linear (left) and logarithmic (right) y-axis. The four-year cycle that corresponds to US election years is prominently visible towards the end of the time period.

Researchers who are interested in obtaining a copy of the current version of the combined data set are encouraged to contact the author. The programs (written in the language Python) used to download, clean and process this data set is also available from the author, while the source of the data set was given in footnote 1 earlier in the paper.

The author plans to continue downloading the source files on a regular basis and thereby improve the temporal characteristics of this data set.

Several publications have already used this data set for research purposes [5,6], while the NC voter data set itself (not the processed and combined version described here) has also been used for experimental research in record linkage [7,8,9]

Researchers who are using this combined temporal data set for experimental or practical research that results in publications are asked to include the following citation in their publication:

Preparation of a real temporal voter data set for record linkage and duplicate detection research
Peter Christen
Technical Report, 2014.
Research School of Computer Science,
The Australian National University
Canberra ACT 0200

The author also appreciates feedback on the use of this data set, especially about data quality issues encountered that might be useful to improving the merging and pre-processing encountered.

Acknowledgements

Thanks to Uwe Draisbach, Brad Malin and Weiyi Xia for conducting consistency checks on the combined data set, thereby preventing that various special cases were pre-processed in a wrong way, and to Uwe for providing the documentation that clarifies the legality of accessing the NCVR database.

References

1. Christen, P.: Data Matching. Data-Centric Systems and Appl. Springer (2012)
2. Herzog, T., Scheuren, F., Winkler, W.: Data quality and record linkage techniques. Springer Verlag (2007)
3. Han, J., Kamber, M., Pei, J.: Data mining: concepts and techniques. 3 edn. Morgan Kaufmann (2011)
4. Naumann, F., Herschel, M.: An introduction to duplicate detection. Volume 3 of Synthesis Lectures on Data Management. Morgan and Claypool Publishers (2010)
5. Christen, P., Gayler, R.W.: Adaptive temporal entity resolution on dynamic databases. In: Advances in Knowledge Discovery and Data Mining, PAKDD, Gold Coast, Australia, Springer Berlin Heidelberg (April 2013) 558–569
6. Ramadan, B., Christen, P., Liang, H., Gayler, R., Hawking, D.: Dynamic similarity-aware inverted indexing for real-time entity resolution. In: International Workshop on Data Mining Applications in Industry and Government (DMApps'13), held at PAKDD'13, Gold Coast, Australia (April 2013)
7. Durham, E.A.: A Framework for Accurate, Efficient Private Record Linkage. PhD thesis, Faculty of the Graduate School of Vanderbilt University, Nashville (2012)
8. Kuzu, M., Kantarcioglu, M., Durham, E., Malin, B.: A constraint satisfaction cryptanalysis of Bloom filters in private record linkage. In: Privacy Enhancing Technologies, Springer (2011) 226–245
9. Kuzu, M., Kantarcioglu, M., Inan, A., Bertino, E., Durham, E., Malin, B.: Efficient privacy-aware record integration. In: ACM EDBT. (2013) 167–178

Appendix: Attributes in the original downloaded NCVR files[3]

Attribute name	Datatype	Length
county_id	smallint	2
county_desc	varchar	15
voter_reg_num	char	12
status_cd	char	2
voter_status_desc	varchar	25
reason_cd	char	2
voter_status_reason_desc	varchar	60
absent_ind	char	1
name_prefx_cd	char	4
last_name	char	25
first_name	char	20
midl_name	char	20
name_sufx_cd	char	3
res_street_address	varchar	63
res_city_desc	varchar	60
state_cd	char	2
zip_code	char	9
mail_addr1	varchar	40
mail_addr2	varchar	40
mail_addr3	varchar	40
mail_addr4	varchar	40
mail_city	varchar	30
mail_state	char	2
mail_zipcode	char	9
full_phone_number	varchar	12
race_code	char	3
ethnic_code	char	3
party_cd	char	3
gender_code	varchar	1
birth_age	int	4
birth_place	char	30
registr_dt	char	10
precinct_abbrv	char	6
precinct_desc	varchar	60
municipality_abbrv	char	6
municipality_desc	varchar	60
ward_abbrv	char	6
ward_desc	varchar	60
cong_dist_abbrv	char	6
super_court_abbrv	char	6
judic_dist_abbrv	char	6

Attribute name	Datatype	Length
nc_senate_abbrv	char	6
nc_house_abbrv	char	6
county_commiss_abbrv	char	6
county_commiss_desc	varchar	60
township_abbrv	char	6
township_desc	varchar	60
school_dist_abbrv	char	6
school_dist_desc	varchar	60
fire_dist_abbrv	char	6
fire_dist_desc	varchar	60
water_dist_abbrv	char	6
water_dist_desc	varchar	60
sewer_dist_abbrv	char	6
sewer_dist_desc	varchar	60
sanit_dist_abbrv	char	6
sanit_dist_desc	varchar	60
rescue_dist_abbrv	char	6
rescue_dist_desc	varchar	60
munic_dist_abbrv	char	6
munic_dist_desc	varchar	60
dist_1_abbrv	char	6
dist_1_desc	varchar	60
dist_2_abbrv	char	6
dist_2_desc	varchar	60
Confidential_ind	char	1
age	int	4
ncid	char	12
vtd_abbrv	char	6
vtd_desc	char	60

[3] ftp://alt.ncsbe.gov/data/ncvhis_ncvoter_data_format.txt