Measuring Completeness and Breadth in Heterogeneous Data

Elena Smith

May 30, 2016

Executive Summary

Radius gains access to the latest business data from its clients' CRM databases, receiving 525,000 new records per hour. While this is a rich trove of information, it must be validated before Radius can update The Network of Record. Radius is careful to protect its database against three vulnerabilities of CRM data:

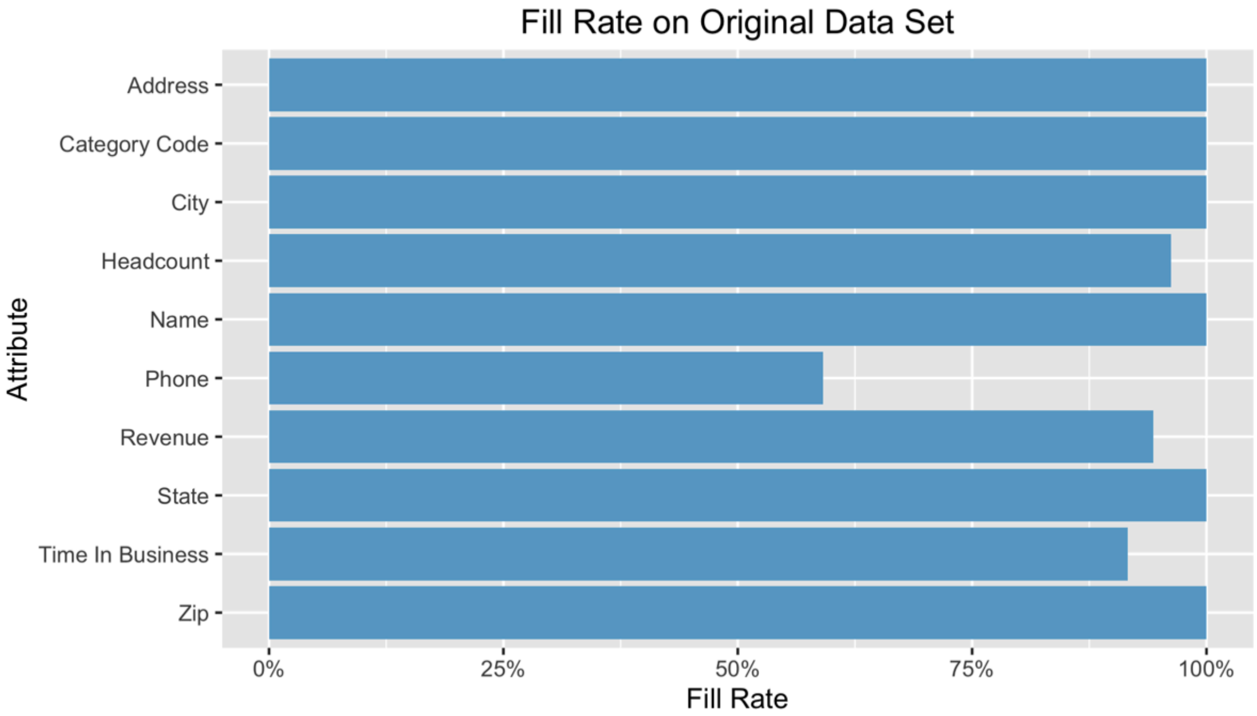
- inconsistent data entry standards across different CRM databases,
- duplicate records,
- and human error in data entry.

In this report, we clean a sample data set with records from heterogeneous CRM databases to address these vulnerabilities. Ultimately, we report on the completeness of the cleaned data set, supplying metrics that help customers benchmark the depth and breadth of Radius's data against data from other providers.

Data Set

The sample data file contains 1 million records and 10 attributes. B2B companies are especially eager for contact information attributes that help them reach clients, such as address, city, state, zip code, and phone number. Other attributes provide context about a business's resources and line of work, such as revenue, head count, NAICS code, and time in business.

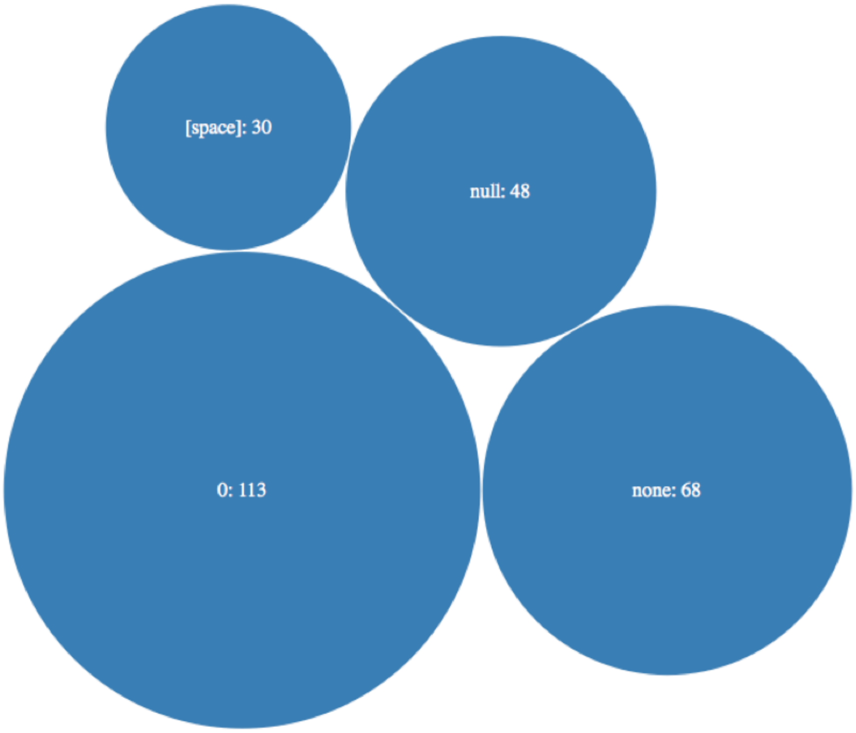
To establish a baseline understanding, we assessed the completeness of the file by measuring the percentage of attributes with non-null values. Six attributes have fill rates exceeding 99% and 3 attributes surpass 90% completeness. Phone number is the only attribute with a fill rate below 90%; 59% of records are populated.



Inconsistent Data Standards Inflate Fill Rates

When data flows in from multiple sources, missing values are rarely marked consistently across sources. The chart above assesses the fill rate by identifying non-null records, but some incoming records employ other coding for missing values. The various encodings of missing values need to be scrubbed out before we can obtain a more accurate fill rate.

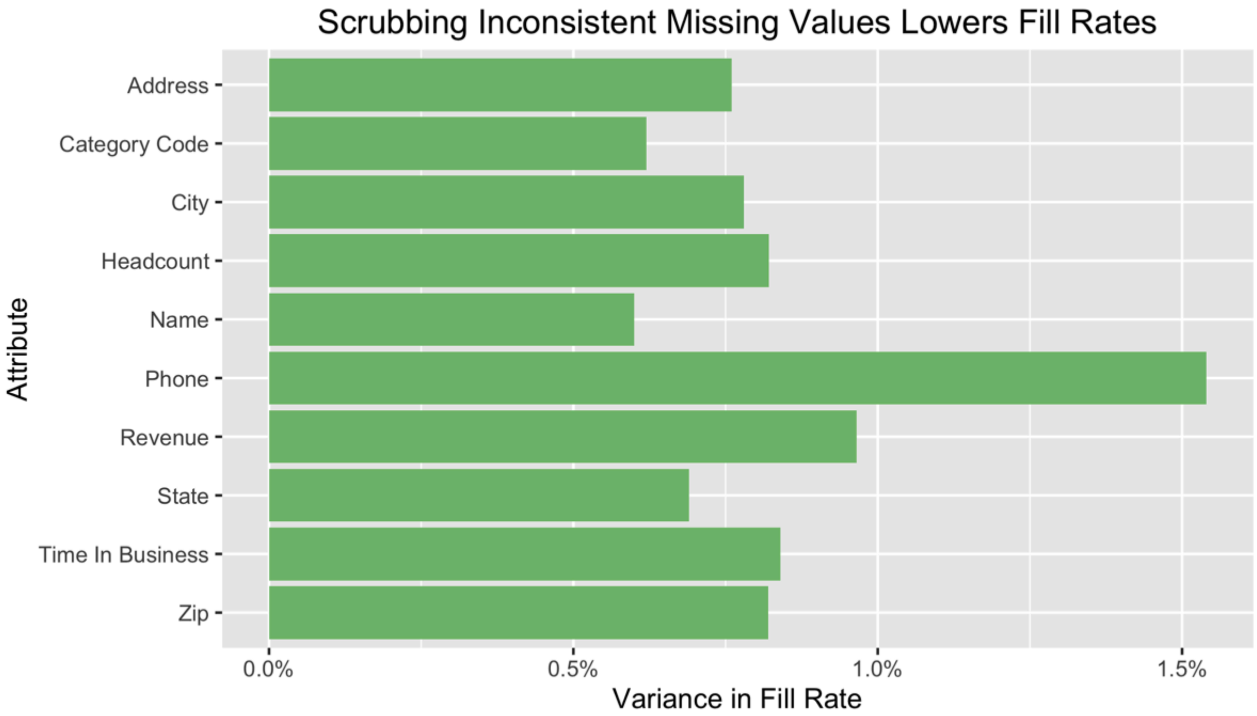
To identify the various missing value codes, we count the frequency of values of the levels of each attribute. For example, if one company uses “null” to indicate that a phone number is missing, the word “null” will most likely appear multiple times in the data set.



The missing value code is indicated on the left, and the frequency of the value across attributes in the data set appears on the right.

The bubble chart above shows four common missing values found in the sample data file, identified by scanning through a list of frequent values. Some databases use the string “null” to code missing values, while others employ “none.” The number 0 serves as a missing value in categorical numeric attributes like phone number.

If we scrub these various missing value codes from the sample file, we obtain a more realistic view of the fill rate of the data file. Performing the missing value scrub reduces the fill rates for each variable by at least 0.5%. Phone number is the most impacted variable, with its fill rate dropping by 1.5%.



Data Entry Errors Boost Fill Rates

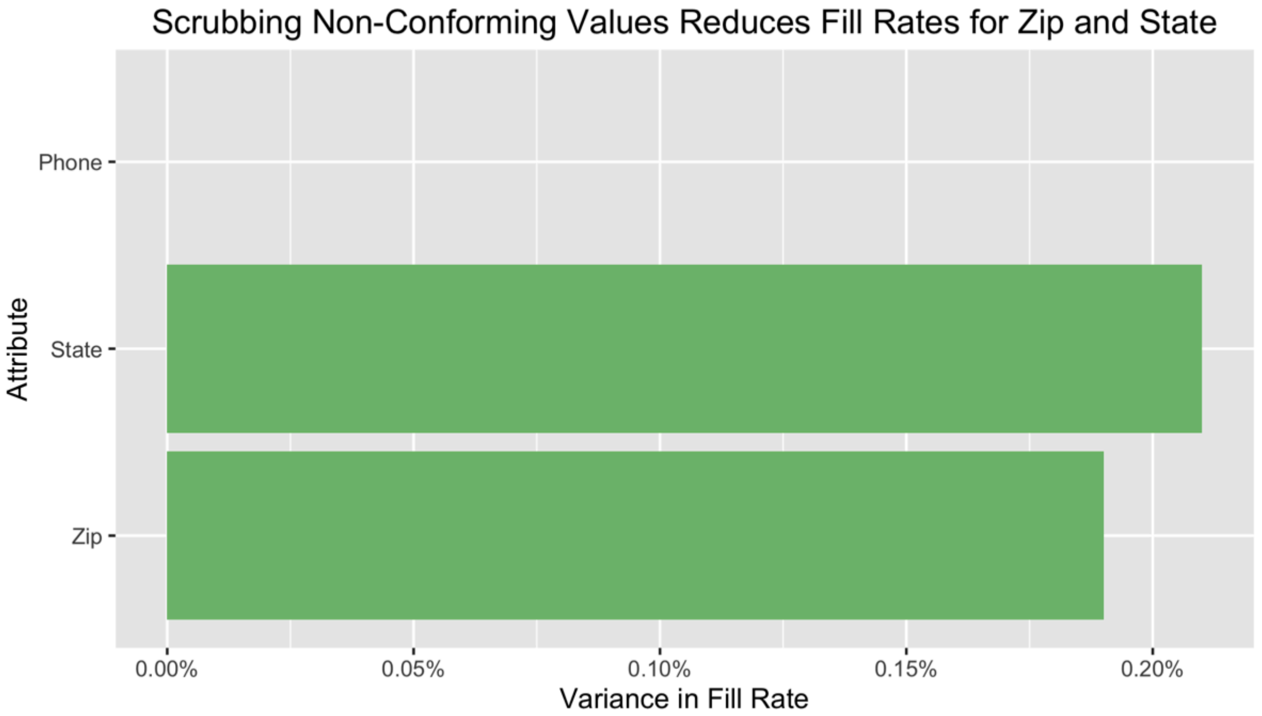
A valid American phone number has ten digits (including the area code), and a valid American zip code has five digits. These are just two examples of data attributes where simple rules can help catch data entry errors. Furthermore, other attributes have a known range of possible values. For example, the United States Post Office publishes a list of valid abbreviations for U.S. states and territories. Since the sample file is limited to American records, state abbreviations that are not on this list can quickly be ruled out as invalid.

We applied the following rules to identify incorrect values and scrub them from the data set:

- **Phone**: 10 digits long
- **Zip**: 5 digits long
- **State**: Only accept abbreviations from a list of two-letter abbreviations for the US states and major terrorities.

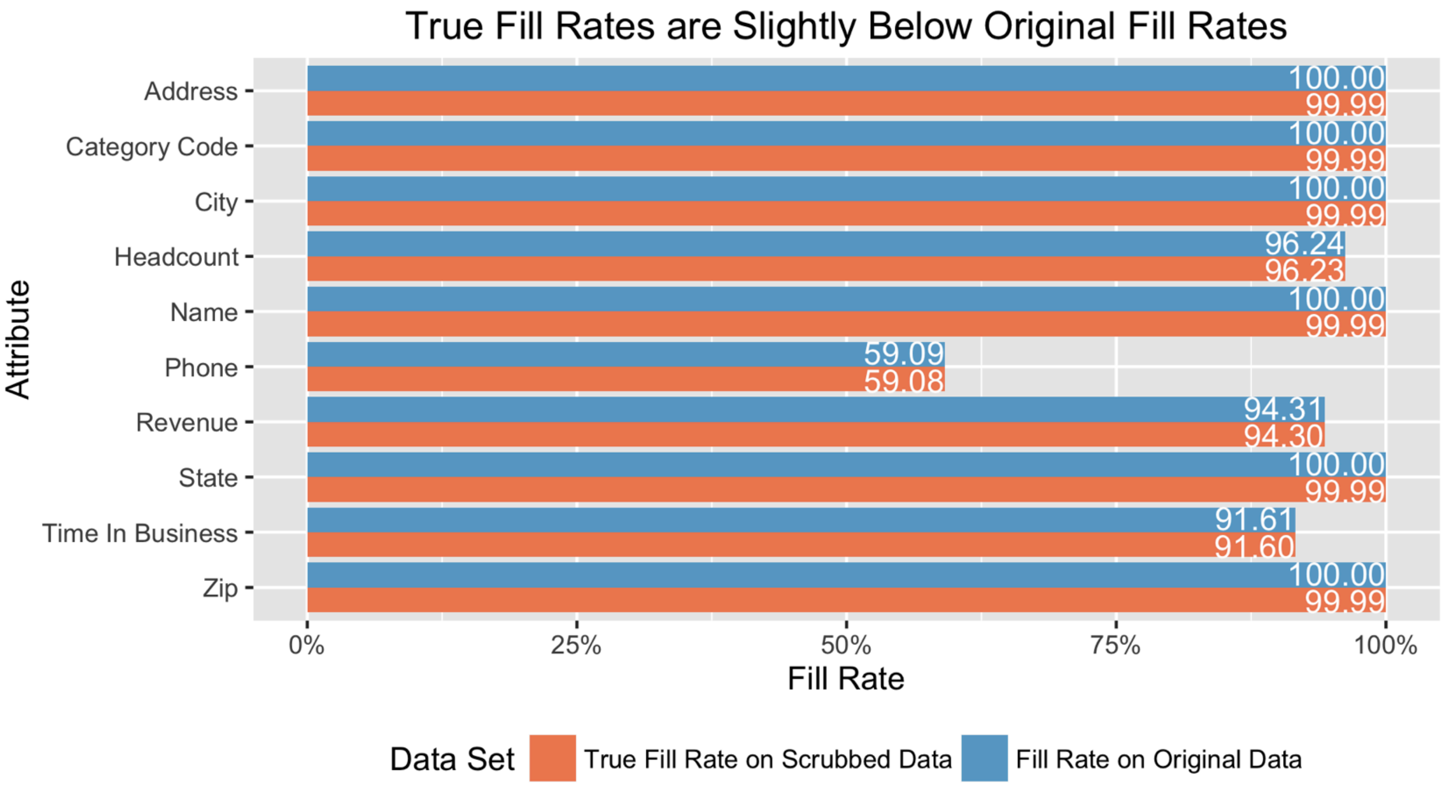
When possible, we took steps to save and correct data that did not meet these rules. For example, 46,532 records have zip codes that are only 4 digits long. However, the states of Connecticut, Massachusetts, Maine, New Hampshire, New Jersey, Rhode Island, and Vermont have zip codes that begin with 0. Excel tends to drop those leading zeroes, so a client who uploads CRM data from an Excel file may inadvertently upload incorrect zip codes. To save this data, we searched for 4 digit zip codes in these states and restored the preceding zeroes.

After scrubbing values that do not conform to our rules, the fill rate for the zip and state attributes dropped by around 0.2%. The fill rate for phones did not drop at all. Since phone numbers encoded as ‘0’ were already removed, all remaining phone numbers in the data set have 10 digits.



Measuring Up: Assessing Completeness and Breadth

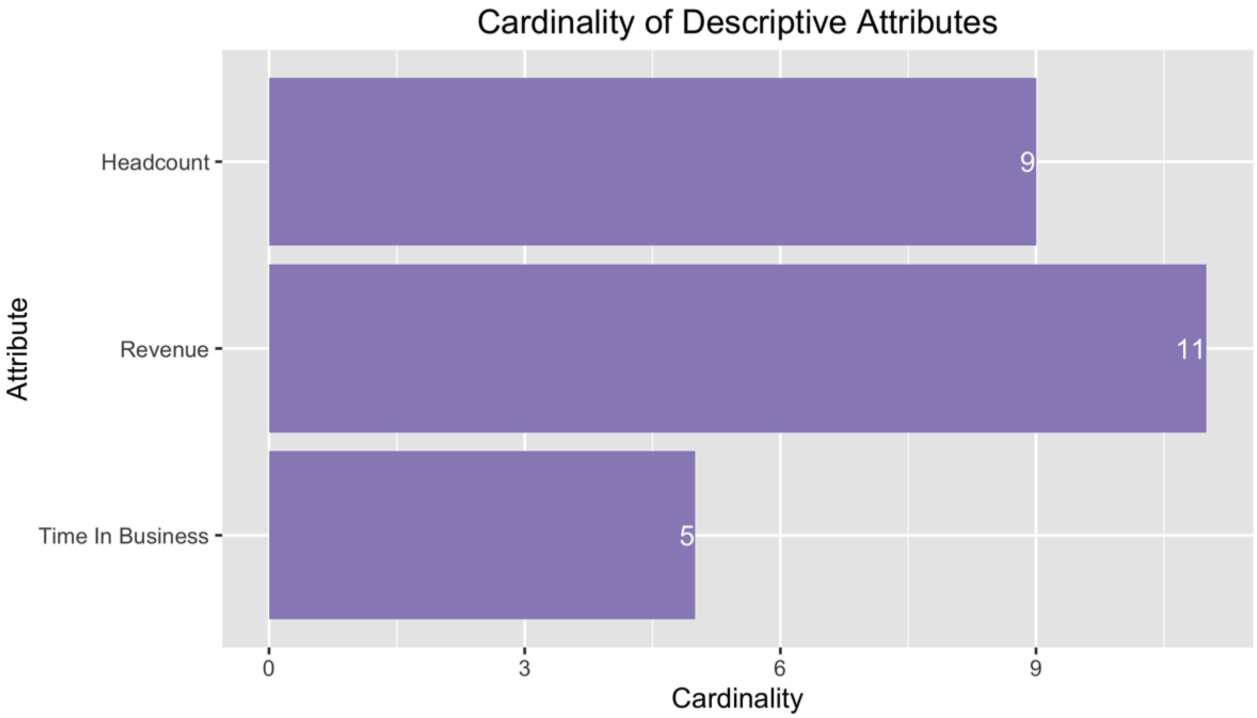
Now that missing and incorrect values have been scrubbed, we can re-assess the fill rate of the data. The true fill rate measures the percentage of “good” values in a data set.



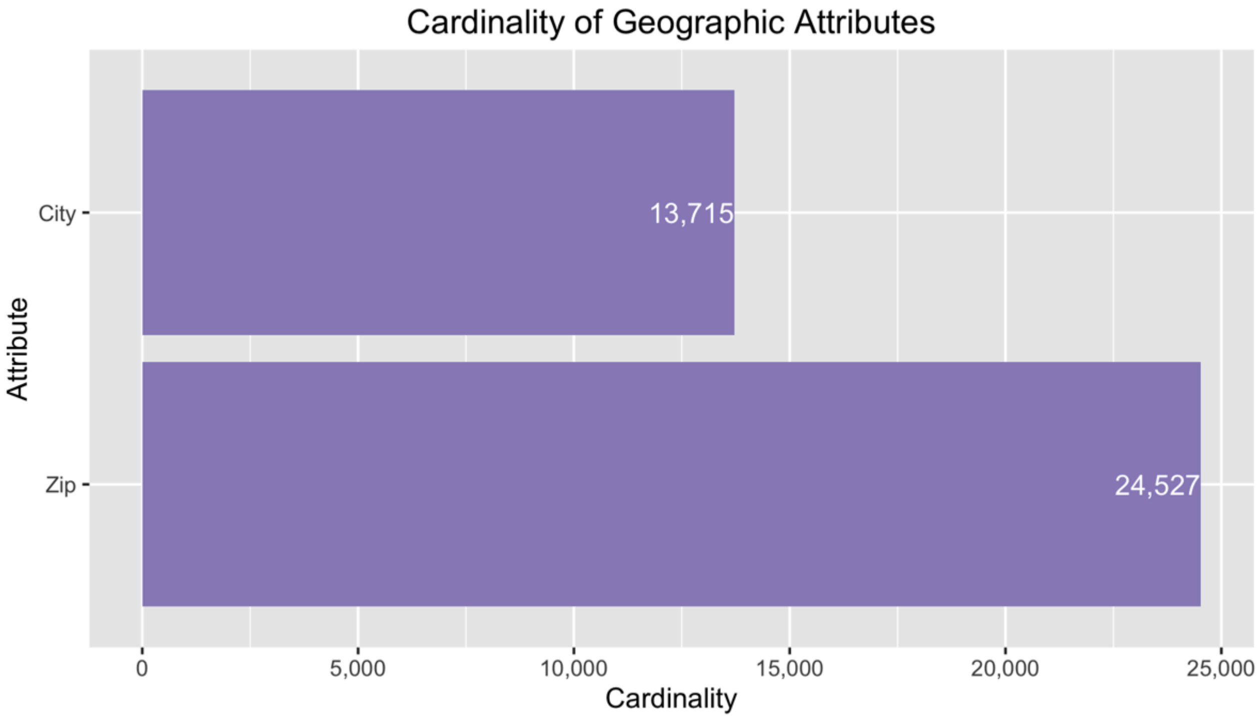
The graph above and Table 2 in the Appendix show that scrubbing the data set of inconsistent missing value codes and data entry errors reduces the fill rate. While the change may be slight, the true fill rate is a more realistic assessment of data quality.

Removing invalid entries from the data set is also useful for assessing the breadth of records in the data set. How many unique records are provided? How detailed is the data? Cardinality is a metric that sheds light on both these questions. Cardinality is a measure of the number of distinct values in a set. In attributes that should be unique, like company name, higher cardinality shows that the file contains records on many different companies. In attributes with contact information, higher cardinality demonstrates that the sample file captures contact data for a large number of companies. In attributes that describe companies, higher cardinality conveys that the sample file has granular information available. A full table of the cardinality values by attribute is provided in Table 1 of the Appendix.

In terms of descriptive attributes, the sample file provides information about companies’ time in business, revenue, and headcount. The cardinalities for these variables are below 15 because buckets are used encode these variables. For example, revenue is grouped into levels like “$5-10 Million” and “$10-20 Million”; precise values are not provided.



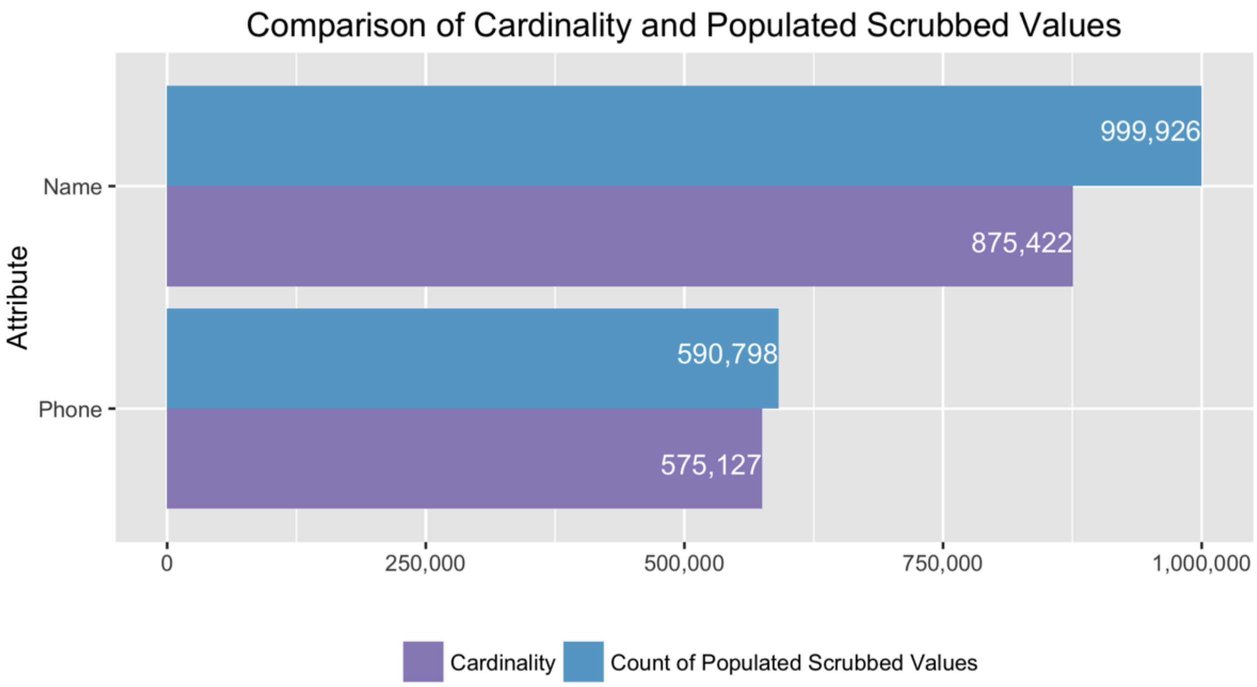
Higher cardinalities for attributes like city, zip code, state, and NAICS code would suggest that the sample file covers a diverse mix of company data. There are approximately 43,000 US zip codes, and the sample file has records on over half of those zip codes. As documented in Table 1 of the Appendix, the sample file also spans 53 distinct states, including all 50 U.S. states, Washington, D.C., Puerto Rico, and the Virgin Islands. Furthermore, NAICS codes are six digit codes that describe business sectors, like Construction and Retail. The sample file features nearly 1,200 different NAIC codes, meaning that a broad array of economic activities are represented.



Cardinality for the name attribute should be high because B2B companies want to access to records on a large number of companies. While the sample file has the name field populated for about 999K records, there are 875K unique business names. Most of the duplicate name values exist because some companies have multiple offices, and distinct addresses appear on separate rows. There are 999,837 unique combinations of address and business name in the data set.

Phone cardinality should also be high because B2B companies desire access to a wider pool of customers. The sample file has populated phone numbers on 590K records, but the cardinality is 575K. Is there duplication of phone numbers because some corporations use a central hotline to direct calls? If so, the number of unique name and phone number concatenated strings should be near the phone cardinality value of 575K. For example, say that 123-456-7000 appears in a 2-record data set twice for ‘Business A.’ The phone number cardinality on this small file is 1, and the cardinality of name and phone number concatenated strings (‘123-456-7000Business A’) is also 1 since the phone number is only linked to 1 business.

However, the sample file contains 590K unique name and phone number strings, but there are only 575K distinct phone numbers. This means that about 15,000 phone numbers are duplicated. These phone numbers are associated with multiple businesses.



One Phone, One Business

2% of the phone numbers in this data set are duplicates linked to businesses with different names, but a salesperson expects to reach one certain company when he dials a specific phone number. Since companies rarely share phone numbers, one phone number linked to multiple businesses signals a data issue. Since Radius accepts data records from a number of CRM databases, phone numbers can become linked to multiple businesses due to data entry errors, inconsistent data entry standards, or data decay.

A simple data entry error in the company name field could make it appear that one phone number is associated with multiple businesses. For example, one CRM database could record a company name as ‘Business A’, while another CRM database mistakenly records the same company’s name as ‘Bsiness A.’ The business names do not match *exactly*, so it appears that the phone number of this business is linked to two different companies. However, the two strings actually represent the same company.

Inconsistent data entry standards can also give a false impression that phones are tied to many different companies. Some CRM clients might create records using the full legal names of businesses, while other clients may store a less formal version of the company’s name (i.e. ‘Lee & Ryan Group’ and ‘Lee & Ryan’). These two entities are the same, but since the strings do not match exactly, duplicate records for the same business entity are created.

Finally, phone numbers may appear multiple times if the data is outdated. When a company closes or moves, its phone number may be recycled and provided to another business. One CRM database may link phone p_1 to the old company, while a more updated CRM database may link p_1 to a newer company.

Understanding the Duplication of Phone Numbers

Can data entry errors or inconsistent data entry standards help explain the presence of duplicate phone numbers in this data set? We analyzed the data to detect misspellings and similar business names.

Levenshtein distances, a metric assessing the number of changes required to make two strings match, can shed light on misspellings in the sample file. The Levenshtein distance of (s_1, s_2) equals the number of insertions, deletions, and substitutions that must be performed to make string s_1 match string s_2. For example, the Levenshtein distance between the set of strings s_a={‘ABC’,‘ABCDE’} is 3 because 3 insertions must be made to make the strings identical.

While a high Levenshtein distance value means that many changes are performed, a low Levenshtein distance value can also indicate a problem. Consider that the Levenshtein distance between the set s_b of strings ‘AB’ and ‘DE’ is 2 because 2 substitutions must be made. However, notice that the two strings do not share any characters; *all* characters are modified. By contrast, the set of strings s_b= {‘ABC’,‘ABCDEF’} also have a Levenshtein distance of 2, but at least these strings share 3 characters in common. In a misspelled word, we expect some characters to be shared between the original word and the corrected one.

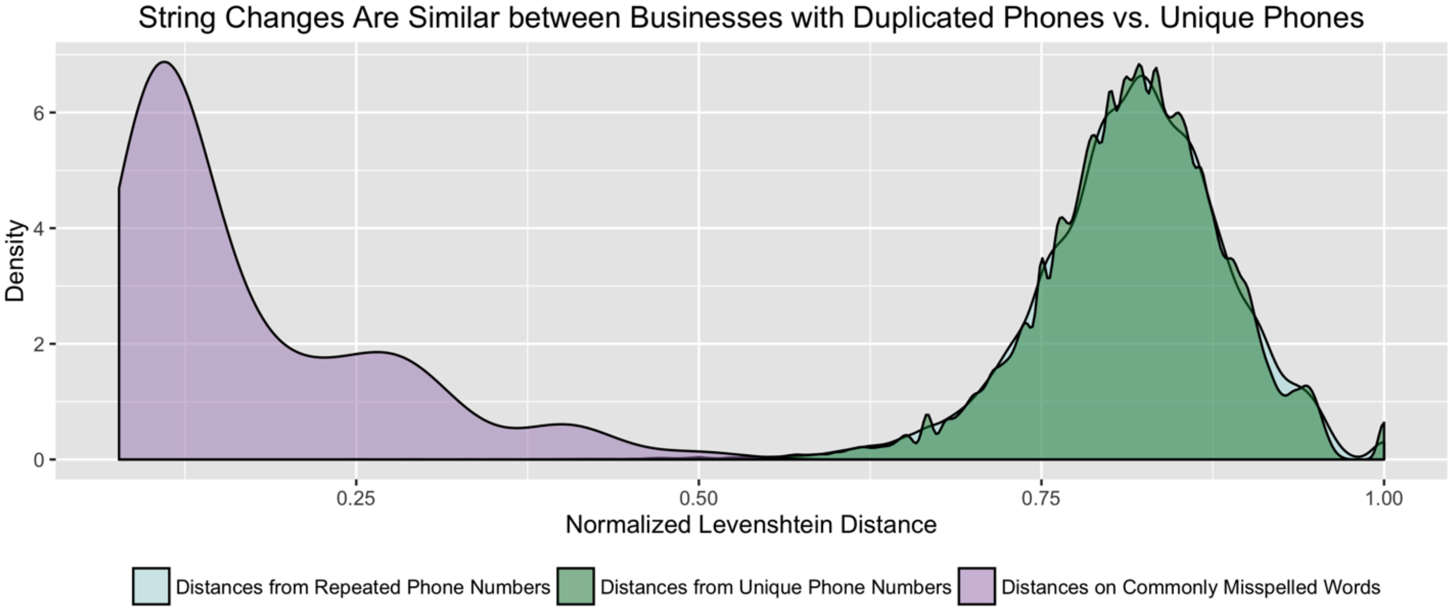
How can we understand the extent to which strings s_1 and s_2 are modified? For this analysis, we utilize the normalized Levenshtein distance, which is equal to the Levenshtein distance divided by the length of the longest string. The normalized Levenshtein distance of ‘ABC’ and ‘ABCDE’ is 2/5 because 2 changes are performed on a string of 5 characters, while the normalized distance of ‘AB’ and ‘DE’ is 1 because 2 changes are performed on a 2 character string. The normalized Levenshtein distance describes the percentage of characters in a string that are changed.

The average normalized Levenshtein distance between two business names associated with the same phone number is 0.82. This means that about 4 out of 5 characters are changed between the names of business 1 and business 2 to make the strings match. Since misspellings typically only involve a few incorrect characters, this sounds high.

But how many changes should we expect? We can benchmark against distinct business names for deeper understanding. Let’s assume that phone numbers that are only associated with one phone number represent distinct and separate businesses. For example, 123-456-7890 is only linked to ‘Lee & Ryan Group’, while 987-

654-3210 is only linked to ‘Farmers Insurance.’ Multiple changes have to be performed to make the strings match because the strings represent two different businesses. Thus, we can assess the normalized Levenshtein distances of distinct businesses with unique phone numbers to understand the typical number of changes required to make the names of two different business match.

We can also benchmark against commonly misspelled words to assess the prevalence of typos in the sample file. The Oxford English Dictionary published a list of 100 commonly misspelled words. We review the normalized Levenshtein distances of the misspelled words and their correct counterparts to understand how many changes are typically required to correct a misspelled word.



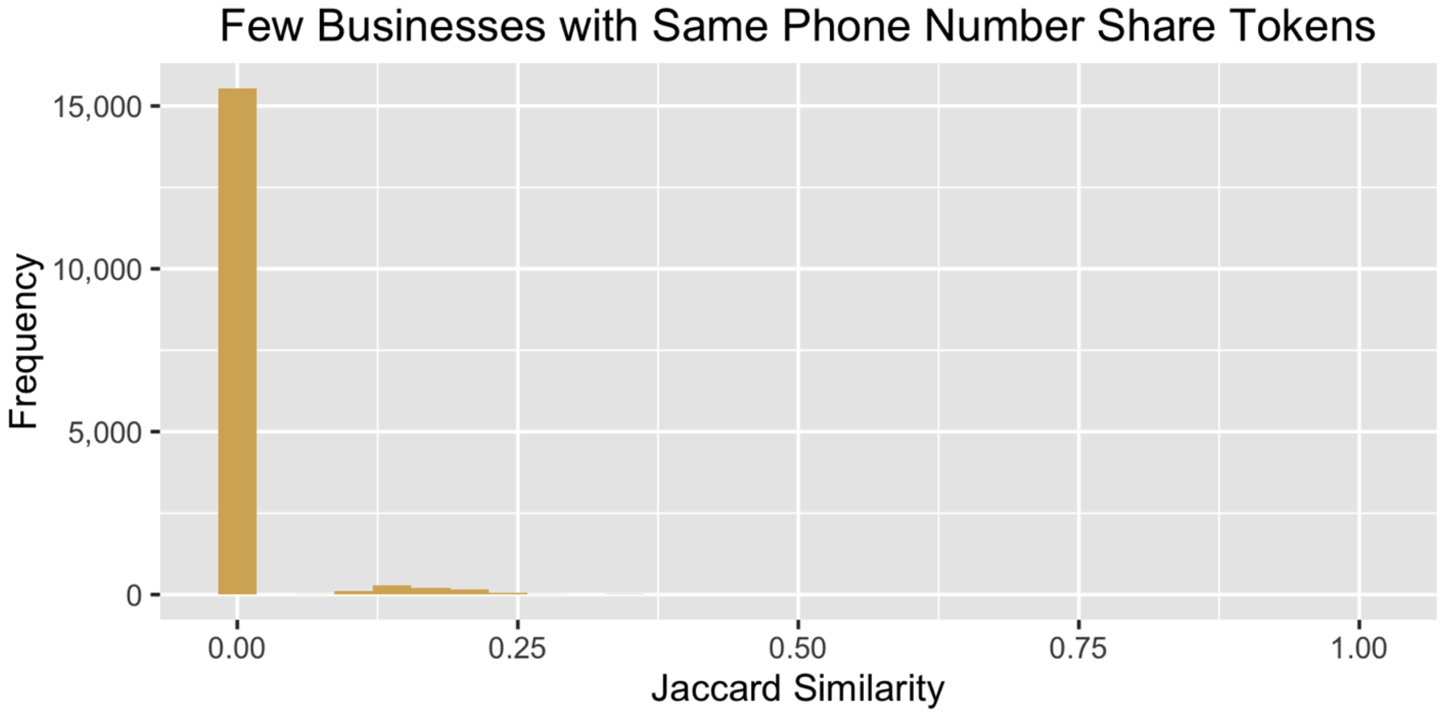
The green-hued areas of the plot above compare the distribution of normalized Levenshtein distances between names of businesses with duplicated phone numbers vs. the names of a random sample of 1,000 businesses with distinct phone numbers. We can see that the distributions are very similar; the green curves overlap to a great extent. For 95% of businesses with distinct phones, 70-90% of characters are changed to make the names of two different businesses match. For 95% of businesses with duplicated phones, 70-90% of characters are also changed. Since the distributions are similar, we can reason that the duplicated phone numbers are generally linked to distinct businesses. Misspellings are most likely not the primary reason that certain phone numbers are repeated in the data set.

For further validation, we can compare the normalized Levenshtein distances of business names against those of commonly misspelled words. The purple-hued portion of the above plot shows the distribution of normalized Levenshtein distances for frequently misspelled words. 10-40% of characters are typically changed to correct a misspelled word. Very few misspelled words require changing 70-90% of characters. We can reasonably conclude that the duplicate phone numbers largely cannot be attributed to misspelled words.

The duplication in phone numbers does not appear to be caused by misspellings, but can it be explained by inconsistent data entry practices? For example, the normalized Levenshtein distance of the set s_c = (‘Lee & Ryan Group’, ‘Lee & Ryan’) is 0.375, but we suspect that these two strings represent the same entity. The strings share 3 words, while one string simply misses the word ‘Group.’

To understand if client databases employ inconsistent data entry practices in recording company names, we measure the Jaccard similarity. This similarity measure describes the number of tokens shared by two strings relative to the total number of unique tokens in the two strings. For example, the Jaccard distance between 'Lee & Ryan Group' and 'Lee & Ryan' is 3/4 because 3 out of 4 tokens are shared. Higher Jaccard distances indicate that more tokens are shared, while lower distances mean that no or few tokens are shared.

The average Jaccard similarity between two business names associated with the same phone number is 0.008. This means that if there were 100 unique tokens between two business names, fewer than 1 in 100 tokens would be shared between the two names. Business names typically only have a few tokens, so this small Jaccard similarity measure shows that businesses with the same phone number generally have no tokens in common. If we assume that a high number of tokens should be shared between strings representing the same business, we can reasonably conclude that the duplicate phone numbers in the sample file are most likely not created due to inconsistent data entry standards in recording business names.



Conclusion

While it may feel like more and more conversations are moving online, phone calls are still the best way to reach prospective clients. In 2015, the Direct Marketing Association found that phone calls have the highest response rate of all contact methods, even higher than direct mail and email.

Thus, maintaining a database of accurate phone numbers is essential to growing and sustaining a healthy sales pipeline. Since we expect one phone number to be associated with one business, we need to resolve a duplication issue: 2% of phone numbers in the sample file are linked to businesses with different names.

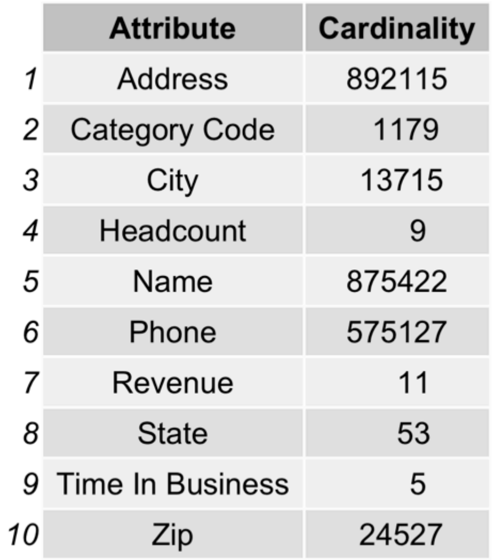
In this report, we analyzed string similarity metrics to determine if spelling errors or inconsistent data entry standards could explain the duplication problem. We could gain a false impression that a phone number is linked to two different businesses if one business appears in the data set twice, once under its correct name and another time under a similar but incorrect name. Ultimately, the business names associated with multiple phones share very few characters and tokens, suggesting that spelling errors and inconsistencies cannot explain the duplication.

Radius reports that 4.2% of phone numbers change or disconnect every 3 months. The practice of recycling phone numbers, or giving a disconnected phone number to a new business, may explain the duplication of phone numbers. Given the high decay rate for phone numbers, we suggest that future analysts study contact records. CRM databases can record each time a person is successfully reached at a specific phone number. Analysts may be able to identify the most up-to-date business for a phone number by studying the frequency and recency of contacts at various phone numbers.

Technical Notes

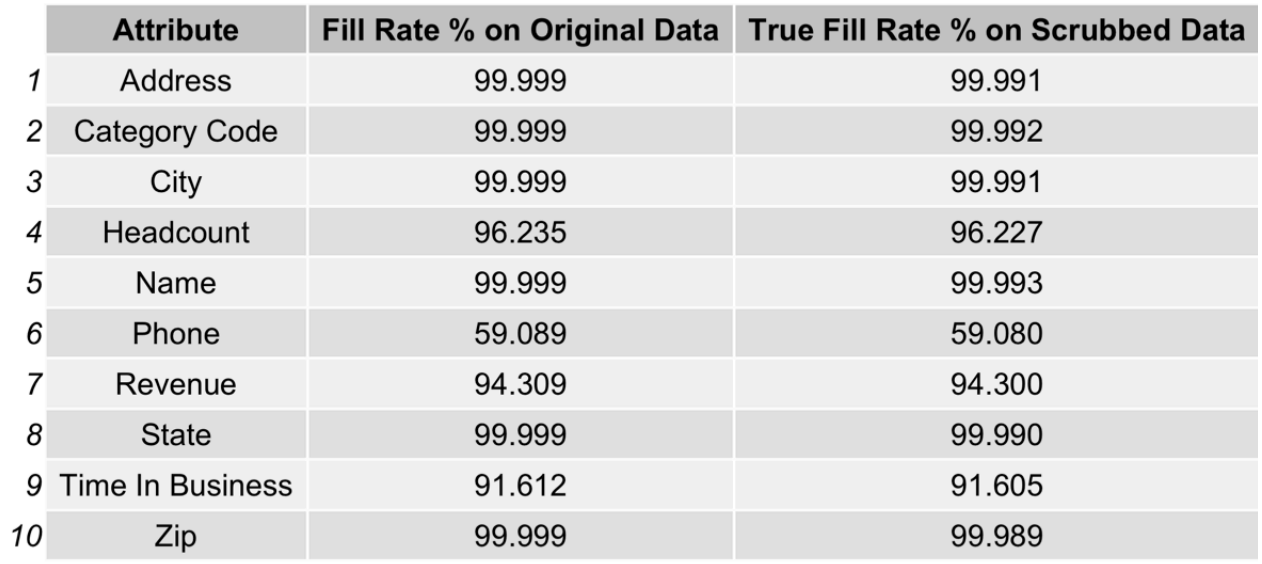
The sample data file is read, scrubbed, and analyzed through the script radiusAnalsis.py (https://github.com/smith-swattie/radius/radiusAnalysis.py). External data on misspelled words is gathered through the script scrapeMisspell.py (https://github.com/smith-swattie/radius/scrapeMisspell.py). Plots for this report are developed in R through the file radiusPlots.R (https://github.com/smith-swattie/radius/radiusPlots.R).

Appendix



	Attribute	Cardinality
1	Address	892115
2	Category Code	1179
3	City	13715
4	Headcount	9
5	Name	875422
6	Phone	575127
7	Revenue	11
8	State	53
9	Time In Business	5
10	Zip	24527

Table 1: Cardinality Values in Scrubbed Data File



	Attribute	Fill Rate % on Original Data	True Fill Rate % on Scrubbed Data
1	Address	99.999	99.991
2	Category Code	99.999	99.992
3	City	99.999	99.991
4	Headcount	96.235	96.227
5	Name	99.999	99.993
6	Phone	59.089	59.080
7	Revenue	94.309	94.300
8	State	99.999	99.990
9	Time In Business	91.612	91.605
10	Zip	99.999	99.989

Table 2: Fill Rates in Original vs. Scrubbed Data Files

References

Cohen, William W., Pradeep Ravikumar, and S. E. Fienberg. “A Comparison of String Distance Metrics for Name-Matching Tasks”. 2003. “http://www.cs.cmu.edu/~wcohen/postscript/ijcai-ws-2003.pdf (http://www.cs.cmu.edu/~wcohen/postscript/ijcai-ws-2003.pdf).”"

Haskel, Debora. “2015 DMA Response Rate Report & Direct Mail.” IWCO Direct Blog, IWCO Direct, 15 Mar. 2017, “www.iwco.com/blog/2015/04/14/dma-response-rate-report-and-direct-mail/.”

Oxford English Dictionary. “Common misspellings.” “https://en.oxforddictionaries.com/spelling/common-misspellings (https://en.oxforddictionaries.com/spelling/common-misspellings)”

Raab Associates, Inc. “How to Solve the B2B Data Quality Crisis: An Expert Assessment of The Network of Record.” “go.radius.com/rs/764-MES-613/images/The-Network-Of-Record-White-Paper.pdf”