GTHA housing market database
OSEMN methodology
Step 2: Scrub (clean) data
Cleanup plan for the Teranet dataset

Stepan Oskin

August 3, 2019

Abstract

Teranet dataset plays an integral part in the proposed GTHA housing market database, design and implementation of which is the primary focus of this Master Thesis. In order to allow the implementation of constraints dictated by the database design, as well as for facilitation of future analysis and modelling based on the database information, it is critical to ensure particular structure and certain degree of consistency of the input data. To address this issue, this document outlines the cleanup plan for the Teranet dataset.

The four main steps of the cleanup process include: spatial join of Teranet points with the polygons of Dissemination Areas, correction of inconsistent entries, addition of new attributes, and removal of duplicated transactions. Additional steps may include: filtering transactions for low and high outliers based on the values of `consideration_amt`, correction of `consideration_amt` for inflation to ensure consistency of dollar values across time, and normalization of values in `consideration_amt`.

1 Introduction

This document presents *Step 2: Scrub* of OSEMN methodology for data science projects. For detailed description of OSEMN methodology, see `methodology/0.osemn/osemn.pdf`.

For background information, description of the Teranet dataset, and its attributes, see `methodology/1.obtain/obtain.pdf`.

As outlined in the document *Step 2: Scrub (clean) data* (see `methodology/2.scrub/scrub.pdf` for details), data analytics, statistical models, and machine learning algorithms, in order to produce meaningful results, require input data to be consistent and transformed into an appropriate form.

In addition to that, as outlined in the *Description of Relational Databases* (see `methodology/rdbms/rdbms.pdf` for details), the main idea of a database is to force the data into a certain structure to allow implementation of constraints and relationships between entities. Organizing data in such a way allows for reduction of redundancy in data storage and drastic improvement in efficiency when working with related datasets. Thus, input data for a relational database also needs to be consistent in order for constraints and relations to work properly.

Data also needs to follow the "tidy" format, as described in section 3 of this document. Teranet dataset is "tidy" in its structure, but does have multiple issues with its consistency, such as inconsistent spelling, missing and erratic values, and duplicate records. This document describes the steps taken to address these issues, improve the quality and usability of the Teranet dataset, and to meet the requirements needed for the organization of the proposed GTHA housing market database via PostgreSQL .

2 Cleanup plan for the Teranet dataset

The four main steps of the cleanup process of the Teranet dataset include:

- Step 2.1: spatial join of Teranet points with the polygons of Dissemination Areas (DAs)
- Step 2.2: correction of inconsistent entries
- Step 2.3: addition of new attributes: `year`, `decade`, and `transaction_id`
- Step 2.4: removal of duplicate records from the Teranet dataset

Additional cleanup steps may include:

- filtering Teranet transactions for low and high outliers based on the values of `consideration_amt`
- correction of `consideration_amt` for inflation to ensure consistency of dollar values across time
- normalization of values for `consideration_amt`

3 Tidy data and database normalization

This section describes the concept of "Tidy Data", as defined by Hadley Wickham. The concept of "Tidy Data" presents the basic ideas of normalization of a database, as defined by Edgar F. Codd, reformulated in statistical language.

3.1 Tidy data

Hadley Wickham in his paper *"Tidy Data"*[1] formalized the way how a shape of the data can be described and what goal should be pursued when formatting data. The principles of tidy data provide a standard way to organize data values within a dataset. The tidy data standard has been designed to facilitate initial exploration and analysis of the data, and to simplify the development of data analysis tools that work well together. The principles of tidy data are closely tied to those of relational databases and Codd's relational algebra[2].

As an integral part of his **relational model**, Codd proposed a process of database normalization, or restructuring of a relational database in accordance with a series of so-called **normal forms** in order to reduce data redundancy and improve data integrity.

3.2 Normalization of a database according to Codd

Normalization entails organizing the columns (attributes) and tables (relations) of a database to ensure that their dependencies are properly enforced by database integrity constraints. As defined by Codd ([2], section 17.5.1), the basic ideas in normalization are to organize the information in a database as follows:

1. Each distinct type of object has a distinct type identifier, which becomes the name of a base relation.

2. Every distinct object of a given type must have an instance identifier that is unique within the object type; this is called its **primary-key** value.

3. Every fact in the database is a fact about the object identified by the primary key.

4. Each such fact contains nothing other than the single-valued immediate properties of the object.

5. Such facts are collected together in a single relation, if they are about objects of the same type. The result is a collection of facts, all of the same type.

3.3 Teranet dataset in the context of a normalized database

Teranet dataset is intended to be used as one of the tables (relations) of the proposed GTHA housing market database that would include other sources of information, such as DA-level demographics, land use information, *etc.* In this context, Codd's basic normalization ideas would take the following form:

1. The Teranet dataset presents a single type of object (relation, or table) —real estate transactions recorded in the Province of Ontario between 1805-01-06 and 2017-10-11.

2. Every distinct object (transaction) must have an instance identifier that is unique within the object type, or its primary-key value. In case of Teranet dataset, all the native columns, including `registration_date`, `pin`, address information, and `X` and `Y` coordinates, have duplicated values present (multiple transactions occurring on the same date, address, coordinates, or under the same pin). Thus, no combination of such columns constitute a candidate key, which prompts the addition of a new attribute that can serve as a unique identifier for Teranet records.

3. A new column `transaction_id` is added to the dataset to be used as its primary key; it corresponds to the row number of each instance (transaction) in the Teranet dataset (filtered to include only GTHA transactions via a spatial join in Step 2.1, see section 4 of this document), ordered from the earliest date to the latest.

4. Every fact in the database is a fact about the object identified by the primary key. This condition is met, as every transaction in Teranet dataset is described by the values found in columns of a single row.

5. Each such fact contains nothing other than the single-valued immediate properties of the object, all columns in Teranet dataset contain single-valued immediate properties of each transaction.

6. Such facts are collected together in a single relation, as they are all objects of the same type (a single table of real estate transactions recorded in Ontario).

Thus, Teranet dataset fits into a normalized database, with the new attribute `transaction_id` as its primary key.

3.4 Codd's constraints formed in statistical language by Wickham

According to Wickham[1], *tidy data* is a standard way of mapping the meaning of a dataset to its structure. A dataset is messy or tidy depending on how rows, columns and tables are matched up with observations, variables and types.

In tidy data:

1. Each variable forms a column.
2. Each observation forms a row.
3. Each type of observational unit forms a table.

This is Codd's 3rd normal form[2], but with the constraints framed in statistical language, and the focus put on a single dataset rather than the many connected datasets common in relational databases. *Messy data* is any other arrangement of the data.

According to Wickham, the most common problems with messy datasets are:

- Column headers are values, not variable names.
- Multiple variables are stored in one column.
- Variables are stored in both rows and columns.
- Multiple types of observational units are stored in the same table.
- A single observational unit is stored in multiple tables.

In the Teranet dataset, none of these problems are present, so it presents *tidy data*.

4 Step 2.1: Spatial join of Teranet points with Dissemination Area polygons

Step 2.1 of the cleaning process of Teranet data involved the spatial join of Teranet points with the polygons of Dissemination Areas (DA). Parameters that were used for the spatial join operation were `how='inner', op='within'`.

The spatial join was performed to filter out Teranet records whose coordinates fall outside of GTHA .

In addition to that, three new attributes were produced as a results of the spatial join:

- Dissemination Area attributes `OBJECTID`, `DAUID`, and `CSDNAME` were added to each Teranet record falling within a particular DA polygon.
- The added attributes allow for extra quality control of Teranet data by comparing the column `MUNICIPALITY` of Teranet records with the column `CSDNAME` associated with DA geometry.
- New columns `OBJECTID` and `DAUID` allow Teranet records to be joined in the future with DA geometry via a regular (non-spatial) join operation (for example, to be aggregated by DAs), or to add any additional DA-level attributes, such as DA-level demographics, to Teranet records.
- These future joins can be performed via regular (non-spatial) join operations, which are much less computationally intensive than a spatial join, and thus can be performed much faster.

5 Step 2.2: Correction of inconsistent entries

Step 2.2 of the cleanup process for Teranet data focused on the correction of inconsistent entries.

5.1 Common problems with data

Common problems with data may include:

- Inconsistent column names
  Column names can have inconsistent capitalizations and/or bad characters.

- Missing data
  Missing data needs to be identified and addressed.
- Outliers
  Outliers can pose a potential problem and need to be investigated.
- Duplicate rows
  Duplicate rows can bias analysis and need to be found and dropped.
- Untidy
  Untidy datasets can contain multiple problems, and prevent us from quickly transforming our dataset from one suitable for reporting to a dataset suitable for analysis.
- Need to process columns
  Columns might need to be processed before they can be used for data analysis.
- Column types can signal unexpected data values
  Column types can signal the presence of unexpected data values.

5.2 Modifications made to Teranet dataset

The following modifications have been made to the Teranet dataset:

1. Inconsistent capitalizations were fixed for:
   - column names
   - column `municipality`
   - column `street_name`
   - column `street_designation`
   - values in the column `postal_code` did not show any problems, but were converted to upper case as a preventive measure
2. Column `province` was removed from the Teranet dataset, as all the transaction are recorded in the Province of Ontario
3. One erratic value was fixed in column `registration_date`

5.2.1 Improper capitalizations

- Column names have inconsistent capitalizations.
- Values in columns `street_name`, `street_designation` and `municipality` have inconsistent capitalizations.
- Values in column `postal_code` do not show problems, but have been converted to upper case as a preventive measure.

5.2.2 Data types

The only column was read with a wrong data type is `registration_date` (read as Python `object`, which in Pandas usually refers to `string`, instead of `datetime`). This happened due to one erratic entry being present in column `registration_date`, which was fixed during Step 2.2 of the cleanup process.

After the erratic value was fixed, `registration_date` can be converted to `date` data type and can be used as one of the indices for Teranet records (however, its values are not unique, as multiple transactions can occur on the same day).

Data types for the rest of the columns are detected appropriately, indicating consistency of data types in the Teranet dataset.

Column `lro_num` can be converted to `category` data type, as it contains only a small range of values. But since this conversion does not result in significant memory savings, its type was left as `integer`.

5.2.3 Missing values

5.2.4 Consistency of entries

1. `lro_num` —consistent. No changes were made.
2. `pin` —non-unique, but consistent. Unique `pin`s might or might not correspond to unique coordinate pairs (same coordinate pair can correspond to multiple unique `pin`s). No changes were made.
3. `consideration_amt` —consistent, but a large number of transactions have very low values of `consideration_amt`, which appear to represent true recorded values (*e.g.,* gift transactions). Also, outliers with very large `consideration_amt` are present in the dataset, which also appear to represent true recorded values and correspond to high-value transactions of commercial property.

4. `registration_date` —consistent, one erratic entry was fixed in Step 2.2.

5. `postal_code` —consistent, converted to upper case as a preventive measure. Non-null values match the correct format for Canadian postal codes.

6. `province` —consistent, removed from the Teranet dataset, as all the transactions are recorded in the Province of Ontario.

7. `unitno` —some erratic values are present, fixed in Step 2.2.

8. `street_name` —some inconsistent values are present (different spellings of street names; with and without street designation; unit/suite number/postal code included; "Highway" or "Hwy"; *etc.*). Correcting all the values presents a time-intensive task, and thus was left for future targeted correction, capitalization were fixed in Step 2.2.

9. `street_designation` —some inconsistent values are present (*e.g.,* "St", "St W", "St.", or "Street"), most common problems were fixed in Step 2.2, capitalization were also fixed in Step 2.2.

10. `street_direction` —some inconsistent values are present (*e.g.,* "S" and "SOUTH"), fixed in Step 2.2.

11. `municipality` —some inconsistent values are present (*e.g.,* "North York" and "North York/Toronto", "Hamilton" and "Hamilton City"), fixed in Step 2.2, capitalization were also fixed in Step 2.2.

12. `street_suffix` —all values are numeric, but consistent, very few values are non-null. As all the values are numeric, it is unclear how do they represent street suffix.

13. `street_number` —consistent, but some of the values are questionably large to represent a street number (*e.g.,* 562916, 47626).

14. `x` and `y` —consistent. Values appear to be in a reasonable range, but some coordinates clearly do not match the address/municipality, where the transaction is supposed to be recorded. However, majority of coordinate pairs appear to be reasonable.

    Records with coordinates that fall outside of the GTHA boundary, as specified by the geometry of the polygons of the GTHA Dissemination Areas, have been filtered out of the Teranet dataset via a spatial join

in Step 2.1. Unique coordinate pairs might or might not correspond to unique `pin`s. There are more unique `pin`s than unique coordinate pairs (same coordinate pair can correspond to multiple unique `pin`s). No changes were made.

6 Step 2.3: Addition of new attributes

Step 2.3 of the cleanup process for Teranet dataset focused on the addition of three more new attributes to each Teranet record (in addition to the three DA-level attributes that were added in Step 2.1).

Two new attributes were parsed from the values of `registration_date` for grouping and visualization purposes:

- `year`
- `decade`

A new attribute `transaction_id` was added to provide unique identifier for Teranet records:

- No combination of native columns provides a convenient unique identification for Teranet records
- `pin`, `registration_date`, `consideration_amt`, `X` and `Y` coordinates, and other columns all have duplicate entries, and thus do not qualify as candidate keys
- A new column `transaction_id` with a simple auto-incrementing range index (row number) will be added to provide a unique identifier for each Teranet record
- The new column `transaction_id` captures the order of transactions in the Teranet dataset, previously filtered to include only transactions with coordinates that fall within the GTHA boundary (see Step 2.1), and ordered from earliest to latest `registration_date`.
- The new column `transaction_id` can be used as a Primary Key for Teranet table within the proposed Teranet database, and thus allows to take advantage of the indexing features available in PostgreSQL

7 Step 2.4: Removal of duplicate records

References

[1] H. Wickham, "Tidy Data," *Journal of Statistical Software*, vol. 59, no. 10, 2014.

[2] E. F. Codd, *The relational model for database management : version 2*. Boston, MA: Addison-Wesley Longman Publishing Co., 1990.