Data Profiling with R

Submitted by Jim Porzak, VP of Analytics, Loyalty Matrix, Inc., San Francisco, California.
JPorzak@LoyaltyMatrix.com

Data profiling should be the first step in any data mining project where we are not 100% certain that the source data actually is what it purports to be -- in other words, always. Done correctly, data profiling will discover data quality issues at the beginning of an analysis project before they impact subsequent processing, or worse, conclusions.

Data profiling is generally considered as part of the data quality process. While data quality is best achieved when, and where, the data is sourced, analytic practitioners don't have the luxury of waiting until a client achieves a state of data quality nirvana. We need to understand the data's limits and deal with it. Data profiling should not be confused with exploratory data analysis (EDA). EDA presupposes valid data. Data profiling discovers invalid or suspect data that must be corrected, discarded, or dealt with in subsequent data hosting and analysis.

A good data profiling tool will

- Require minimum input from an analyst to run.
- Do column profiling with simple statistics, plots, patterns, exceptions and domain detection.
- Do dependency profiling to identify any intra-table dependencies impacting normalization.
- Do redundancy profiling to discover keys between tables and other overlapping variables.
- Produce easy to use output which will be useful to analysts and meaningful to clients.
- Save findings to an accessible data structure for subsequent use.

Unfortunately commercial data profiling tools are generally targeted at "enterprise" data integration projects. They tend to be highly flexible and thus complex to use. Of course, they come with an enterprise worthy price tag.

At Loyalty Matrix, we must deliver data-driven insights quickly and economically. One of our tricks is to profile all new data sets upon receipt. Our data profiling tool has evolved over the last year having been used on dozens of projects.

Since our clients send us gigabyte, or low terabyte, datasets, we routinely load all client data into a SQL relational database (RDBMS), typically MS-SQL or MySQL. The initial step in any project is loading the raw data without transformation into the RDBMS. Our profiling tool takes advantage of this by using a combination of R and SQL (via RODBC) to optimize processing time and flexibility. We use grid graphics to create data summary panels that can be displayed individually or combined into a comprehensive report. Summary statistics and data hypothesis are written back to the RDBMS for subsequent reporting and integration into metadata repositories.

This paper will

- Review our requirements for the data profiling tool.
- Describe the high level design and structure of the tool.
- Show examples of the integration of R and SQL to achieve optimum processing.
- Show examples of grid graphics design and code for the data summary panels.
- Conclude with real-world examples of quality issues discovered with help of the tool.

end