DQ Description

We implemented these tests using R, a high level technical computing language which is designed for data analysis, statistics, some data wrangling, visualization and some machine learning. The reports are using a particular library called knitr, which takes markdown combined with R code to produce the html files that you probably saw. Markdown is a more or less well-defined standard for marking up text such that it can be transformed to a variety of outputs (docx, html, pdf, etc), it is used fairly extensively and you probably would recognize it as the markup that Reddit or Wikipedia us in their editors.

The rules are implemented in an R script. We actually keep the R script separate from the markdown skeleton, because we continue growing our battery of tests as we learn through the execution of our audits. The tests are fairly straightforward and they are based on each variable’s class or type, with a few statistics produced for all variables:

* General (applies for all classes)
  + Observation count
  + Count of distinct observations
  + Count of missing values (NA is for R what NULL is for SQL)
  + Count of blank values (useful for strings that are a only spaces)
* Character/strings
  + Descriptive statistics for character length (we do have a preference for robust stats, but we include mean and standard deviation as well)
  + Top 25 frequent values
  + Identification of strings that include non-letter/number characters (! , . ? \*…)
  + Identification of strings that have 3 or more repeated characters
* Numeric
  + Descriptive stats
  + Number of outliers using Tukey’s standard definition (Q25-1.5 IQR, Q75 +1.5 IQR), we also include a count of +/- Inf which is useful when someone creatively divides by 0 or takes a log of a negative number
* Factor
  + Factors is a data type in R that is useful to represent labels, such as currencies or business lines (they are a little more than strings and this data type is much more efficient than strings)
  + We give a count of the most frequent values and cutoff at 25 elvels
* Boolean
  + We give counts of values
* Dates
  + WE give a subset of descriptive stats
  + We show the oldest and most recent dates

We modify them according to the domain we are working on. For instance, we may check validity of SIN numbers, or verify that birthdates do not imply the existence of 120-year olds. Ultimately, we work with the auditors to determine what makes sense to check.

I am unsure if you are looking for additional information to get a better understanding of the finding or to generate ideas for your own implementation. If it is the later, I would strongly recommend that you do not use R. This is a great prototyping tool, but it is not necessarily great for production-strength. If we needed to do this on a regular basis for the same data sets or a data feed, I would probably go with Python or look into an off-the-shelf product. You may also want to talk to the Data Quality experts at EDMO, who have a fairly robust solution.