Testing Data Completeness with DQe-c-v2

OHDSI Symposium 2019: Data Quality Workshop
09/17/19

Tim Bergquist, Graduate Research Assistant
Biomedical Informatics & Medical Education
University of Washington
WWAMI region Practice & Research Network

- 60+ Primary care WWAMI clinics
- ~20 data connected clinics
- CHCs and RHCs
- Underserved populations
- Many serving rural populations
- Collaboration with national network of practice based research networks
- Data QUEST represents over 250,000 patients

https://dataquest.iths.org/
Data QUEST

- 20 data-connected clinics in the WPRN
- Represents over 250,000 patients

An electronic health data-sharing architecture across community-based primary care practices in the WPRN
Measuring Data Quality Framework

Operationalizing the framework into: 5 conceptual tests and 17 discrete tests across:

- **Completeness**
  - Are the data present?

- **Conformance**
  - Are the data standardized and formatted?

- **Plausibility**
  - Are the data believable?


Measuring Data Quality Framework

Operationalizing the framework into: 5 conceptual tests and 17 discrete tests across:

- **Completeness**
  - Are the data present?

- **Conformance**
  - Are the data standardized and formatted?

- **Plausibility**
  - Are the data believable?


Measuring Data Quality Framework

Operationalizing the framework into: 5 conceptual tests and 17 discrete tests across:

- Completeness • Are the data present?
- Conformance • Are the data standardized and formatted?
- Plausibility • Are the data believable?

# Data Quality Tests

<table>
<thead>
<tr>
<th>DQ Framework category</th>
<th>TEST</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>COMPLETENESS</strong></td>
<td>Gender, Visit, Observation completeness (denominator and proportion with valid data)</td>
</tr>
<tr>
<td><strong>COMPLETENESS</strong></td>
<td>Key clinical status completeness (denominator and proportion with valid data): Smoking status, alcohol consumption</td>
</tr>
<tr>
<td><strong>COMPLETENESS</strong></td>
<td>Measurement completeness (denominator and proportion with valid data): Height, Weight, SBP, DBP</td>
</tr>
<tr>
<td><strong>COMPLETENESS</strong></td>
<td>Cross reference tables that are present in current dataset to expected tables in standard OMOP CDM</td>
</tr>
<tr>
<td><strong>COMPLETENESS</strong></td>
<td>Looks for NULL and invalid variable values in each column and visualizes percent missingness</td>
</tr>
<tr>
<td><strong>CONFORMANCE</strong></td>
<td>Check that primary and foreign keys relate properly; High Priority: Person_ID, Visit_Occurrence_ID</td>
</tr>
<tr>
<td><strong>CONFORMANCE</strong></td>
<td>Checks that orphan don't keys exist (a foreign key is present in a table but no primary key exists in the reference table)</td>
</tr>
<tr>
<td><strong>PLAUSIBILITY</strong></td>
<td>Comparison of new load to old load (Number of observations, Number of unique patients, Number of tables with rows)</td>
</tr>
<tr>
<td><strong>PLAUSIBILITY</strong></td>
<td>Size of tables and rows across the OMOP CDM</td>
</tr>
</tbody>
</table>
Original DQe-c Tool

Modular tool developed in R for assessing **completeness** in EHR data repositories. Customization and configuration was difficult

- Hard to add new modules
- Difficult to add new CDMs (or new versions of CDMs)
DQe-c-v2 Tool

Modular tool developed in python for assessing **completeness** in EHR data repositories.
DQe-c-v2 Tool

Takes in the database credentials, CDM version, and configurations.
DQe-c-v2 Tool

Takes in the database credentials, CDM version, and configurations.

Simply enter your credentials and configurations into the config.json file.
DQE-c-v2 Tool

Takes in the database credentials, CDM version, and configurations.

Simply enter your credentials and configurations into the config.json file.

Run:
python DQe-c.py -c /path/to/config.json
DQe-c-v2 Tool

Sets up the database connection, manages report output, and initiates the CDM files
DQe-c-v2 Tool

Assesses conformance to a Common Data Model. Checks for missing tables and calculates size of tables.
Figure 1. Available Tables, Compared to all CDM (OMOPV5_0) Tables

This figure shows which of the CDM tables are loaded and/or available.

```
## Warning in `\$.data.table`(dtfDT, , `:`=(`c", fact), with = FALSE):
## with=FALSE ignored, it isn't needed when using `:`. See `?`:=' for examples.
```
Figure 2. File Size and Row Numbers by Table in the (OMOPV5_0) Load

Size represents number of rows and color represent file size (in GB) for each table.
Quickly check that the new data is growing as expected
DQe-c-v2 Tool

Assesses **completeness** of all columns in the available tables in the database. Checks for null and nonsense values.
Identify empty or useful columns in each of your OMOP tables.
DQe-c-v2 Tool

Checks for orphan keys, foreign keys not present in the primary table.
DQe-c-v2 Tool

Checks for missingness in clinical indicators. (What percent of patients have a heart rate measure, blood pressure measurement, etc.)
Figure 5. Common Key Variables

Figure 5 shows the percentage of patients missing specific key clinical indicators.
Adding a new indicator test is straightforward!

**Completion as the presence of a concept.**
Calculates what percentage of patients have the identified concept(s).

```json
{
    "indicator name": "heart rate",
    "table": "MEASUREMENT",
    "col": "measurement_concept_id",
    "label": "HR",
    "concepts": [4239408]
}
```

**Completion as the presence of a non-null.**
Calculates what percentage of patients have a non-null value in the identified table-column.

```json
{
    "indicator name": "Medications",
    "table": "drug_exposure",
    "col": "drug_exposure_id",
    "label": "Medication",
    "concepts": false
}
```
We can add a new indicator test by just adding five new fields.

Adding testing for A1C Hemoglobin. Calculates what percentage of patients have a hemoglobin A1C measurement.
Test of Completeness in Key Clinical Indicators

Figure 5. Common Key Variables

Figure 5 shows the percentage of patients missing specific key clinical indicators.
DQe-c-v2 Tool

All reports are combined into a visualization dashboard
DQe-c-v2 Tool

All these modules output csv reports. The output folders are managed by Query.py
DQe-c-v2 Tool

All these modules output csv reports. The output folders are managed by Query.py to account for different test dates and organizations.
DQe-c-v2 Network Aggregation Tool

A network-level preview of table availability

<table>
<thead>
<tr>
<th>attribute_definition</th>
<th>cohort_attribute</th>
<th>concept_class</th>
<th>condition_occurrence</th>
<th>domain</th>
<th>drug_exposure</th>
<th>measurement</th>
<th>note</th>
<th>observation</th>
<th>observation_period</th>
</tr>
</thead>
<tbody>
<tr>
<td>care_site</td>
<td>cohort_definition</td>
<td>concept_relationship</td>
<td>death</td>
<td>dose_era</td>
<td>drug_strength</td>
<td>payer_plan_period</td>
<td>procedure_occurrence</td>
<td>provider</td>
<td>relationship</td>
</tr>
<tr>
<td>cdm_source</td>
<td>concept</td>
<td>concept_synonym</td>
<td>device_cost</td>
<td>drug_cost</td>
<td>fact_relationship</td>
<td>person</td>
<td>source_to_concept_map</td>
<td>visit_cost</td>
<td>visit_occurrence</td>
</tr>
<tr>
<td>cohort</td>
<td>concept_ancestor</td>
<td>condition_era</td>
<td>device_exposure</td>
<td>drug_era</td>
<td>location</td>
<td>procedure_cost</td>
<td>specimen</td>
<td>vocabulary</td>
<td></td>
</tr>
</tbody>
</table>
DQe-c-v2 Network Aggregation Tool

Network-wide changes in the main clinical tables by site and across data reload

This is an aggregate view of the primary loads across tables for the entire network. This allows a comparison.
DQe-c-v2 Network Aggregation Tool

Network-wide missingness in available tables.
Figure 9. Indicator differences across the network

Figure 9 shows the different indicator measurements from across the network.
DQe-c-v2 Tool

Reports are visualized into an HTML file. Easy to embed into a website.
Adding New Modules

class Example:
    def __init__(self, query):
        self.query = query

    def runTest(self):
        # write your script here

        # If you have SQL queries make sure to accomodate the different query structures
        # at the end you should have some pandas dataframe with statistics
        final_output_report = some_pandas_dataframe

        # write your report to the current report folder with the query function outputReport
        self.query.outputReport(final_output_report, "output.csv")
Vocabulary Summary

Figure 6. Vocabularys in Use by Clinical Table

Figure 6 shows the percentage of all concepts in the clinical tables. The tests are derived from the tests/vocabulary.json files.
Vocabulary Summary

Ratio of vocabularys in use in the table

Table: condition_occurrence, death, drug_exposure, measurement, observation, procedure_occurrence

Percent of Terms: 0, 25, 50, 75, 100

Vocabulary:
- HCPCS
- LOINC
- RxNorm
- SNOMED
Figure 6. Vocabularys in Use by Clinical Table

Figure 6 shows the percentage of all concepts in the clinical tables. The tests are derived from the tests/vocabulary.json files.
Figure 7. Changes in Record Numbers across Time

Figure 7 shows the number of records over time in the repository.

Temporal Plausibility
Operationalizing use of DQe tools for data quality testing

* Data QUEST
* DARTNet Institute
* CD2H

https://github.com/WWAMI-DataQuest/DQe-c_OMOPv4/tree/master/docs
Questions?

• We are looking for collaborators and contributors!
• Contact me if you need help getting the tool up and running.
• We are always looking for feedback.

Thanks to Kari Stephens, Hossein Estiri, WPRN, ITHS, and CD2H!

Contact: Tim Bergquist trberg@uw.edu

https://dataquest.iths.org/

https://ctsa.ncats.nih.gov/cd2h/

https://github.com/data2health/DQe-c-v2

CD2H Data Quality Project