# Cleaning Data Using Structured Query Language (SQL)

By Farrokh Alemi version Tuesday, May 30, 2017

Edited by Nancy Freeborne on February 3rd 2017

## Data errors: Dead People Who Show for Visits

Before the data are merged from different files, it is important to exclude records of patients that are impossible. For example, sometimes a patient is reported to have a visit after death. Clearly this is not possible. In rare situations some visits occur after death—for instance, the transport of a dead patient from home to hospital may be possible. Other scenarios can also exist where the patient has an encounter after death. These are rare and very specific to post mortality services. The most common reason for encounters with health care system after death is due to date of death being entered wrong by a clinician or clerk in the healthcare system. This is usually not the case, if we are working with Center for Medicaid and Medicare Services (CMS) Death Master List; but in recent years CMS has decided against sharing this master list because of potential for identity theft. Therefore, we may be left with an erroneous date of death. One of the first steps in cleaning the data is to identify patients whose date of death occurs before date of various outpatient or inpatient encounters. The code for identifying such discrepancies may look as follows:

SELECT id

INTO Exclude

FROM Dx

WHERE [Age at Death]<[Age at First Dx]

GROUP BY id;

In this code, the “SELECT id” command tells the system that we are interested in finding the ID of the patients. Note that we do not need to identify the source of the ID field as this query has only one table, so there is no confusion where the variable comes from. It is always best to specify so there is no room for confusion. Thus we should have used: “SELECT Dx.id”. The INTO command says that we should include these IDs into a file called Exclude. The “FROM Dx” command says that we want to get this information from a table called Dx, which incidentally includes both age at death and age at diagnosis. The “WHERE [Age at Death]<[Age at Dx]” command says that age at death must be less than age at diagnosis. The “GROUP BY id” says that we want to see only one value for each ID no matter how many times the person’s diagnosis occurs at a later age than the age of his death. Also note that the WHERE command will be executed before the GROUP BY command.

## Data Errors: Patients with No Visits

In many studies, we are looking for encounters with the healthcare system for a patient. Sometimes the entries in a medical record are not for real people, as when a test case was entered. These patients are typically identified typically with primary keys that start with ZZZ. These records must be excluded before proceeding.

For some patients, during the study period there are no encounters with the health care system. This creates doubt about whether these patients are real or simply healthy. If the period is long, say a decade then some encounters with the healthcare system is expected. In the absence of any encounter, it is important to explore why this may be occurring. For example, the patient may be using the facilities for picking up his/her medications but not for receiving medical services. Other explanations are also possible. It is important to count how many patients have no encounters with the system and find out what is the most likely explanation of this finding.

## Data Errors: Imputing Missing Values

Missing values are another common error. Keep in mind that the medical record is the report of an encounter. Patients may have encounters that are not reported as when the patient meets the clinician in a social gathering. A question arises regarding what should be done with information not reported in the electronic health record. The answer depends on what the information is. For example, let us think through what should be done when the patient has no report of having diabetes. One possibility is that the patient was not diagnosed with diabetes, in which case we can assume that the patient does not have diabetes. In this case, we will set the value of the field diagnosis to 0:

IIF(Diabetes1 is null, 0, 1) AS Diabetes2

This command says that if the field Diabetes1 is null, replace it with 0 and otherwise assume it is 1. Rename the new field as Diabetes2.

One general strategy for imputing missing values is to assume that missing values are the most common value. Thus, if the diagnosis of diabetes is missing, if most patients in our study do not have diabetes diagnosis, then the best approach may be to assume that the patient does not have it. However, we cannot always assume that a missing diagnosis indicates normal condition. In the emergency room, where there may not be time to diagnose the patient a missing diagnosis may indicate insufficient time to establish it and may indicate that serious conditions were present. In one study, for example, in an emergency room, we found that missing values of myocardial Infarction (heart attack) diagnosis were highly correlated with mortality. In this situation, it is not right to assume that missing diagnoses indicate normal condition.

Treatment is usually reported when given. It may be safe to assume if treatment is not reported that it was not given. Again it may be important to understand if patient conditions precluded giving the treatment.

Sometimes when a value is missing, the best approach is to measure it from its nearest values. For example, if the blood pressure is missing in a particular quarter it may make sense to estimate it from the next most recent time period. Other times we can impute the missing value from available data. Thus, we can impute that the patient is diabetic, if he/she is taking diabetic medication or if hemoglobin A1c levels (a marker for diabetes) indicate diabetes. Exceptions occur, especially if medications are used to impute diagnosis. Some pre-diabetic patients take Metformin, a drug also used for diabetes, so therefore indicating that those patients had true diabetes would be erroneous. Physicians may prescribe a medication for a reason different than the typical use, , so if we see the medicine in the database, we may think it indicates a diagnosis when it does not. In these situations, we would need to understand if the rest of the patient’s record supports the imputation.

## Data Errors: Inconsistent Data

One way to have more confidence in data in medical record is to find out if the data are consistent. A patient whose age shows as more than 110 years may have an erroneous date of birth. Here is a sample SQL code to detect if age is above 110 years old:

SELECT Id, [Age at Encounter]

FROM Encounter

WHERE [Age at Encounter]>110;

A pregnant male is not be a reasonable record. Here is a code to detect if male subjects are pregnant:

SELECT Id

FROM Table

WHERE Gender = “Male” and Pregnant is True;

Of course, inconsistent data does not arise only with impossible combinations. Some level of inconsistency may arise among variables that often co-occur. Very tall people may be unlikely to be in our sample even if their height is possible. If we see a body weight of more than 400 lbs, we wonder if the patient’s weight was taken while he/she was on a motorized chair. Even among probable events, inconsistent data should raise concerns about quality of the data. When all fields point to the same conclusion, then there is little concern with the quality of the data. When some fields suggest the absence of an event and other fields suggest that the field has occurred then a concern is raised, often requiring human chart reviews or a conversation with the patient. An example may demonstrate.

The Agency for Healthcare Research and Quality (AHRQ) has come up with several measures of quality of care using electronic health records. One such measure is the frequency with which medication errors occur. When a medication error occurs, clinicians are required to indicate it in the record. However, sometimes this is not done. Sometimes, an activity is done but not recorded in the right place. Thus, AHRQ’s patient safety indicator may rely on frequency with which heart failure patients are discharged with beta blocker prescriptions (an evidence-based treatment). The doctor may have put the prescription in the note but since the patient was going to their children’s home in a different State, they may have filled the prescription using a pharmacy that was not monitored in our medication reconciliation files. In these situations we would under count the number of beta blockers. The variation in reporting is one reason AHRQ recommends that expensive chart reviews be done to verify under or over reporting of patient safety issues. Some chart reviews, however, are not needed if other indicators are consistent with reported event.

To see if other variables in the electronic health record are consistent with the reported event, we predict the event from other variables. Then, comparison of predicted and observed values indicates the extent to which data are consistent. For example, one would expect that patients who are older, who had long hospital stays, who have multiple medications, and who have cognitive impairments are more likely to fall. Suppose we have developed an index, or plan to use an index found in the literature, that predicts patients who are likely to fall. Then, we can compare the predicted probability of fall to actual observed fall. If there is negligible probability of fall and the patient has fallen, then something is not right. If there is high probability of fall and we see consequences of fall (prolonged hospitalization), then maybe the patient has fallen and the information is not correct inside the EHR. Table 5 shows hypothetical results. Chart reviews may need to be done in the cases of low probability events that were reported and high probability events that were not reported.

**Table 5: Reports Inconsistent with Probability of the Event**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Probability of Fall | | |
| Low | Medium | High |
| Fall Reported | Not consistent |  | Consistent |
| Fall not reported | Consistent |  | Not consistent |

A similar test of consistency can be applied to patient-reported outcomes such as pain levels. Obviously, pain is a subjective symptom. Some patients have more tolerance for pain than others. Other patients report the same level of pain but some decide to treat it with medications and others refuse medication. One can predict the expected pain level from the patient’s medical history. Then expected and reported pain levels can be contrasted. When patient-reported pain levels do not fit the patient’s medical history, then additional steps can be taken to understand why that is the case. For example, the patients’ medical history can be used to predict the potential for medication abuse. If the patient is at risk for medication abuse and is reporting inconsistent pain levels, then the clinician can be alerted to the problem so the clinician can explore the underlying reasons.

**Table 6: Expected and Patient Reported Pain Levels**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | Expected Pain Level | | |
| Low | Medium | High |
| Patient Reported levels | Low | Consistent |  | Not Consistent |
| Medium |  |  |  |
| High | Not Consistent |  | Consistent |