Intake and Process

Summary

This paper is meant to cover simpler situations with data. I'm not talking here about high-performance data pipelines that read huge quantities of data in real time, dynamically handle it, and send it to other processes. That requires people smarter than me to talk about. I'm focusing on the day-to-day intake of something someone sends and someone else needs to make something useful out of it.

Definition

To me, an intake and process is a fancy way of designing and building a method to plan for the receipt of data, apply standardized processes to it, and produce documented, usable data. I specifically write 'receipt' because I want you to think about it as a first two-way exchange (request from someone/thing and receive from someone/thing).

Common situations where it applies

One-off research – you receive data files for a paper you are writing on a subject you understand.

Clinical Research Projects – you receive data multiple times during a clinical research trial and need to prepare periodic reports and analyses.

Data Products – you receive data and prepare on some periodic basis datasets for a consumption by others.

I'm going to describe most of this based on the work I did for NEHRI, with some details/concepts from my time at Westat. NEHRI was an especially good learning experience for me, because before that point, I'd never dealt with meteorological and pollution data. I didn't understand the structure, the terminology, the usefulness of the fields or how people used the data. I'd also never done spatial analysis, spatial joins, or coordinate systems.

Clarifying roles, timelines, moving parts

I earlier wrote "receipt," and I want to expand on that here. At a minimum you are told you are getting data. A better (and less frequent) situation is to be part of the process where you ask what data formats, fields, and transmission methods are available. Of course, with enough time, you can write something to read in data with ambiguous types for each column, or clean up extraneous rows, or remove unprintable characters. And oftentimes you must. But just having an upfront conversation: "What options do you have for exporting?" can lead to hours of saved labor and reduced headaches.

Getting data via Excel is not the best method, unless the creator has formally set the formats for each column to ensure Excel knows which columns are text, which are dates, which are numbers, which are, whatever. Also, that the creator has made a 'tidy' sheet where there are no merged cells, or extraneous rows, or carriage returns in cells, or unprintable characters.

It starts with identifying who is sending the data and having even just a brief conversation with them as to what, and how, and when. The transfer process should keep date/time stamps on files, or the receipted files should be uniquely identified with date/timestamp.

Prepare to be wrong

How many times have you ordered take-out, and something is missing, or not what you ordered? Same with data receipts. The first one is always a learning experience. So just be prepared.

Receipting

Unless automated, use a folder-based system to separate 'batches' of incoming data files. If you are doing a process where you need to communicate/make available status to others, then keep a document where you log in the data that is accessible by others. Sometimes you want to record not only receipt, but if there are delays, files missing, or other issues.

Initial Analysis

Using a standardized program, you want to be able to quickly read in the data files, and completely document what you have. A summary report also helps by giving you a 'first page' report with the list of files, date, observation count, field count, record length, overall information. If you receive the files periodically, it is good to have a QC report for each data extraction to compare with the prior data receipt. The report should highlight files that are missing or added, or instances when the number of fields has changed. Even observation differences should be checked, although with periodic receipts, you really want to focus on unusual changes (such as files that have less observations by some %). In some cases, studies drop records for extraction (e.g., a patient withdraws), and while always good to confirm, it often doesn't mean something is wrong.

Additional reports should completely document the structure, statistics, frequencies, and values of the fields. Again, if this is a process that receives the data frequently, you may want to flag instances where a column has more missing values as a percent than it did before, or where the type of a column doesn't match the type receipted before.

Now the fun part, you must look at everything. If there aren't labels, create them. For each variable, look at the distribution if numerical, for categorical, look at all the values. Are there typos? Are there mixed case differences? Does anything look odd, is really the mantra to follow. Get clarification/confirmation on anything you see, including rules for how to manage them. Also, check to see if there are paired fields. For example, a field that contains a value, and an associated field that contains the measurement of the other field (e.g., "feet," "inches," "cm," ...) Those pairs should be check together to ensure the values make sense with the associated measurement. I've seen a 150 feet baby (in the data) because the measurement was feet, but the value was in centimeters. This sometimes helps the data owner realize they have issues that need to be addressed before you get the next data submission.

If the subject matter is new to you, then do additional research looking up the meanings of fields you aren't familiar with. This will help you find out the submitted data fields are typos.

Rename, Recode, Revise

Now that you know as much as you can about the data, you plan what renames, recodes and revisions to make. I often add an 'OrigOrder' column to each data set so that even if I rename fields, or create new key fields, or sort multiple ways, I can always match up later versions with the original data.

Renaming should be done carefully by doing a 'pre-post' rename check. Recoding values is the same. Keep the original field on the file long enough to do a 'pre-post' recode check. I think it is good practice to have an automated routine for doing recodes based on meta-data and automatically report out the work for inspection. There is a "No Code Recode" presentation written that delves into this method.

Revising structure is also important aspect to consider. You have gotten one file, but after coming to understand it (or what you need to use it for), you may decide to split it. You may get multiple files and decide to split and combine a few. In a perfect world, we are all at least 3rd, 4th, or 5th normal form [1]. I've yet to see huge benefits on a small clinical trial for normalizing data that far. If you've made it so you can analyze it correctly, you've done well. Big caveat: if one of the requirements is that you are submitting this data to a regulatory organization (such as the FDA), then that file structure you send will be entirely determined by the organization and may in fact require very specific standards (such as CDISC, HL7, FHIR, etc.)

Finished product

Now that you've resolved everything, fixed everything, made it look just like you want, put it into a nice, finished product. I refer to this as a data 'package.' Package in the non-technical sense. It is a box, with a nice cover, and inside you'll find all sorts of gifts, like a packing slip, a user manual, and everything nicely named and organized. You also want to put a copy of the package someplace safe, as backup in case the recipient of your package accidentally let's their pet chew on it.

User Documentation

For deliverables to others, a memo (that packing slip) is a wonderful way to provide a summary of the package and explain how the pieces fit together. It can provide background on why the package was created and how to use it. It can highlight issues with the data or special handling needs (especially with weighted data). It can identify who to contact if there are questions.

The dataset documentation is the user manual. As with the initial state, it should detail each field. If there are known issues, create reports showing the issues (cross-referenced with the memo). I prefer electronic documentation over printouts because of searchability, environmental concerns, and because certain things don't print well (like an interactive zoom in/out of a map showing spatial information). If you have structured, book-marked, searchable documentation, don't worry about page count.

Even if the documentation is for a one-off research paper, it should be complete enough so that if you need to refer to it in a few years, you aren't wishing you had done a better documentation job.

Planning for time

Lastly, for data that gets updated regularly, be prepared for the unexpected. The intake process will hopefully highlight new tables, fields, or data values. Effectively integrating them takes serious thought and you want to do this before it happens. I'm not kidding when I say, this is hard to do, especially if you are not a content expert. When I first started working with NEHRI data, I only heard of "environmental data" and "monitoring stations." I treated them as the root concept of organizing data. The model has held for four years, but I have had to use "monitoring stations" specific to meteorological and pollutant monitoring stations from two national agencies. As other sources of pollutants became data sources, I split them by source (Crops, Industries, Quarries, Water, ...). While splitting meteorological and air pollution up front would have been possible, it would have made processing messy (both agencies provide both types of data in the same files).

So, the structure is still acceptable for now, and I try to be more careful when a new data source comes along to see if it fits or breaks the organizational model.

[1] https://www.studytonight.com/dbms/database-normalization.php