Lessons learned from building a database with colleagues
#NICAR19
MaryJo Webster, Minneapolis Star Tribune, @MaryJoWebster
Todd Wallack, Boston Globe, @twallack. twallack@globe.com
Dana Amihere, KPCC, @write_this_way

TOOLS:
You will have the most success if you use a tool that provides you the ability to restrict what gets entered and provides the greatest efficiency. For example, having pre-selected options in a pull-down menu or restricting the format (only a date can be entered in a date field). You may also want to restrict editing/deleting access to just one person to prevent unintentional changes. Alternatively, consider restricting editing access to portions of the data.

Google Forms is a free option. You just need one Gmail account to set it up in. It allows you to create a web page for entering data, and the data is then stored in a Google Sheet. Limitations: Can't edit records from the form view (need to go into the back-end spreadsheet). With a little coding knowledge you can use the "pre-filled link" option to send out a partially-filled form to specific people.

Airtable is a more advanced option. It can be used for free for up to 2,000 records. Beyond that you will need to buy a subscription (roughly $10 per user, per month for up to 5,000 records). Airtable has more options than Google Forms for restricting what gets entered into a field, and also restricting users. It's also much easier to search and edit existing records. More about Airtable here.

Custom built tools -- some news organizations, such as the Washington Post, have used Django or other programming languages to build their own data entry forms, hosted on a newsroom server.

Open Semantic Search: This is an open source software that you would have to install on your newsroom server. It allows you to organize and "tag" PDF documents. For the Star Tribune project, this proved crucial for keeping track of which documents had been read by a team member and which had not so that we didn't duplicate our work, or miss something. Setting this up requires some technical know-how, however.

Overview: The same organizing and tagging features of Open Semantic are available in Overview, which is a free, online tool created by data journalist Jonathan Stray (and others). Once you upload your documents into a set, then you can tag and annotate documents. It's also possible to essentially create a simple spreadsheet and export it. Multiple people can work at the same time. Limitation: you can't restrict what people type in the fields. However, you could solely use it for organizing and tagging documents in the same way that the Star Tribune used Open Semantic (no technical know-how or server required)

Mechanical Turk or hiring a data entry firm -- Mechanical Turk is a platform that allows you to outsource tiny tasks to workers, often for a few pennies each. For instance, you might ask workers to go to a website, solve a captcha, and copy one piece of information. ProPublica wrote an excellent introduction to the service in 2010.

BEFORE YOU GET STARTED:

--What is(are) the project goal(s)?
--Convene a team meeting to discuss goals, expectations, etc.(if you're working alone, bounce your ideas off an editor or colleague)
--Don't just start typing! You need to think carefully about what fields you are going to include and what exactly will be typed into those fields. Once you start entering records, you don't want to have to go back and do it again because you realized you made a mistake! (Or worse, having to redo an interview or get documents you failed to get the first time.)
--Think ahead about what you're going to do with your data -- Are you going to join it with other data? If so, do you have the right field/values needed for joining?
--Ask yourself what you might want to say in your story as a way to figure out what fields you might need.
-- Make sure all the staffers are involved in the discussion (including reporters, editors, and graphics).
--Seek expert guidance regarding what fields to include or how to track something, especially in situations where you might be making judgment calls.
--Keep a backup copy of the source documents in an alternate location (ideally on a server that gets backed up on a regular basis). If it's small enough, set up a free Dropbox account and stash it in there. Or upload it to a Google folder.
--What is each row/record in the dataset going to represent? Sometimes this might be trickier than you think. For example, if your data is about crime, are you tracking the incident, the victim(s), or the suspect(s)? If it's victims or suspects, you might need more than one record for each incident.
--What fields are ones you must have in the database? Which ones would be nice to have? Which ones do you think are unnecessary? You want to find a balance between a strong dataset and one that is manageable within your time and manpower resources.
--Each column should only have one value.
--Set up the data entry form using pre-selected choices for as many of the fields as possible. Set fields as "required" for ones that can't be skipped.
--Avoid including too many "free form" entry fields, since those can slow down the data entry process and are hard to use for statistics. However, you may want a place for reporters to write notes that are pertinent to that record.
-- Be prepared for needing to list a value as "unknown." It's better to type something in the field (such as "N/A" or "unknown") rather than leaving it blank. You (or your co-workers) will have a hard time discerning that a blank field means you weren't able to get that information versus you simply forgot to fill it in.

--Make sure to record the document (and/or page number) that the data came from in case you need to go back and compare your data to the original source. In some cases, that could be a URL for the webpage you used.
--Include the name of the person who entered the information
-- Instead of using spaces in your column names, considering underscores, hyphens, or SnakeCase. And try to be consistent in your style. (Don't use ALLCAPS for one column, for instance, and all lowercase for another.)
--Type a dozen or so records into your data entry form, then stop and assess whether it's working the way you had hoped. If needed, make changes and go back to fix the records already entered.

HOW TO ENSURE TEAM CONSISTENCY:
--Make a cheat sheet to share with all team members, detailing how each field should be filled out and answering key questions.
--Have a plan for divvying up the work, so that you don't duplicate work or end up missing things.
--Set aside time to ask each other questions to ensure you're on the same page, especially in the first few days of doing data entry.
--Avoid moving, renaming or copying your original documents (or whatever is the source you are building your dataset from). Except that you should keep one backup copy, as noted above.
--Consider limiting access to the back-end data to avoid unintentional deletions or changes
--Run some basic analysis on your data after you've entered some data, but not too much. You might find data-entry problems that can be fixed, or you might discover you're missing a crucial piece

SHARING YOUR RESULTS

Markdown pages in R can be a good way to share your results
Example: http://strib-data-public.s3-us-west-1.amazonaws.com/projects/rape/highlights.html
--If you're not using R, come up with a systematic way to share that avoid sending "updated" versions (i.e. a shared directory where team members, editors, etc can view the latest documents and old versions are overwritten). You don't want multiple versions floating around in numerous locations, which might yield confusion or even result in someone using the wrong information.
-- Make sure everyone understands what the results mean.

Projects we talked about:

MaryJo: Denied Justice project where we had a team of 4 people reading thousands of police reports on sexual assaults and then filling out dozens of fields in a data entry form -- some of the information was taken directly from the report (such as the date it was reported) but others required us to make judgments based on narratives (such as whether the police interviewed all potential witnesses). www.startribune.com/deniedjustice

Todd. The Valedictorian Project. My colleagues tracked down more than 150 former valedictorians from Boston area schools to find out how they fared in college and their careers. They entered information from their interviews into Google Sheets, including a couple dozen fields. The project was successful, but could have been a lot easier if everyone involved had come to this panel.
https://apps.bostonglobe.com/magazine/graphics/2019/01/17/valedictorians/

Todd. Private schools, painful secrets. My colleagues and I at the Boston Globe created a database of allegations against staffers at more than 100 private schools in New England. We initially tracked down allegations at 67 schools. Then we created a Google form to compile tips from readers, which helped us eventually expand the database to 110 schools.
https://www.bostonglobe.com/metro/2016/05/06/private-schools-painful-secrets/OaRI9PFpRnCTJxCzko5hkN/story.html

Dana. Race for the Oscars. I did some quick analysis for a colleague at The Dallas Morning News for a story on Oscars diversity, which later turned into an entire database and accompanying data write-up. For the second year in a row, the Oscar nominees in the four acting categories (best actor and actress in a lead role, best actor and actress in a supporting role) didn't include a single minority. I set out to answer a simple question with a visual database: Has diversity always been this problematic in the most prized acting categories?
http://interactives.dallasnews.com/2016/oscars-diversity/

USA Today Network: 80 reporters around the country spent two weeks searching yearbooks for racist photos. https://www.usatoday.com/in-depth/news/investigations/2019/02/20/blackface-racist-photos-yearbooks-colleges-kkk-lynching-mockery-fraternities-black-70-s-80-s/2858921002/