

Case Study: Death By Numbers

By Hernán Adasme, Dan Howlett and Emily Meyers

Introduction	1
About the Project	1
About the Data	3
Overview of Transformation Process	3
Metadata Creation, Omeka Import	3
Dataset Creation with DataScribe	5
Transcription with DataScribe	7
Transcription Forms	7
DataScribe for Project Management	8
Workflow	8
Transcribers	9
Reviewers	10
DataScribe for Research	11
Dataset Export	11
Conclusion	11

Introduction

Death By Numbers is a project to transcribe and analyze the London Bills of Mortality, a weekly and annual series of mortality statistics produced from 1603 to 1752. With almost 8,000 bills from the weekly series alone, there is quite a bit of data to sift through and transform into a manageable format for computational analysis. In this case study we will explain how Datascribe was an instrumental part of this process of transcription and transformation.

About the Project

[Death By Numbers](#) is a project of the [Roy Rosenzweig Center for History and New Media](#) and the major goal of the project is to transcribe and publicly publish the information in these bills in a dataset suitable for computational analysis. This project looks at one of the most dreaded diseases in early modern England, the plague which was present in the British Isles from 1348 until 1679. The most well-documented epidemics of the early modern era were in England's cities, particularly London, which suffered six major epidemics in the century between 1563 and 1665, and lost an estimated 225,000 people to plague. Government officials attempted to quantify the severity of various plague outbreaks and, starting in 1603, published London's weekly mortality statistics in broadside series known as the Bills of Mortality. The bills grew to include not just plague deaths but also dozens of other causes of death, such as childbirth, measles, syphilis, and suicide, ensuring their continued publication for decades after the final outbreak of plague in England. The weekly bills were also supplemented annually with a general account of the preceding year, published on the Thursday before Christmas. Between 1603 and 1752, almost 8,000 different weekly bills were published, chronicling plague and general mortality through the city of London.

A general Bill for this present year, ending the 19 of December 1665. according to the Report made to the KINGS most Excellent Majesty.

By the Company of Parish Clerks of London, &c.

Buried	Plu.	Buried	Plu.	Buried	Plu.	Buried	Plu.
S ^t A'bens Woodstreet	200	121	S ^t Clements Eastcheap	38	20	S ^t Margaret Mofes	38
S ^t A'hallowes Barking	514	336	S ^t Dunns Back-church	78	27	S ^t Margaret Newfish	114
S ^t A'hallowes Breadst	35	6	S ^t Dunfons East	265	150	S ^t Margaret Pattons	49
S ^t A'hallowes Great	455	420	S ^t Edmunds Lombard	70	30	S ^t Mary Abchurch	99
S ^t A'hallowes Honils	10	5	S ^t Ethelborough	195	106	S ^t Mary Aldermanbury	181
S ^t A'hallowes Lette	239	175	S ^t Faiths	104	70	S ^t Mary Aldetmary	105
S ^t A'hallowes Staining	185	52	S ^t Follers	144	105	S ^t Mary le Bow	64
S ^t A'hallowes the Wall	500	356	S ^t Gabriel Fen-church	69	39	S ^t Mary Bothaw	55
S ^t Alphage	271	115	S ^t George Botolphiane	41	27	S ^t Mary Colechurch	17
S ^t Andrew Hubbard	71	25	S ^t Gregories by Pauls	376	232	S ^t Mary Hill	94
S ^t Andrew Underflath	274	189	S ^t He'kens	108	75	S ^t Mary Mounthaw	56
S ^t Andrew Wardrobe	476	302	S ^t James Dukes place	262	190	S ^t Mary Summer set	342
S ^t Anne Alder gate	132	197	S ^t James Garlickhithe	189	118	S ^t Mary Staynings	47
S ^t Anne Blacke-Friers	652	467	S ^t John Baptist	138	83	S ^t Mary Woolchurch	65
S ^t Antholms Parifi	58	22	S ^t John Euangelist	9	5	S ^t Mary Woolth	75
S ^t A'ufins Parifi	43	20	S ^t John Zacharie	85	54	S ^t Martins Iremouger	21
S ^t Barthol. Exchange	73	51	S ^t Katherine Coleman	299	213	S ^t Martins Ludgate	196
S ^t Bennet Fyoch	47	23	S ^t Katherine Creechu	335	231	S ^t Martins Orgars	110
S ^t Benn. Grace-church	41	21	S ^t Lawrence Iewry	94	48	S ^t Martins Outwitch	60
S ^t Bennet Pauls Wharf	355	172	S ^t Lawrence Pountney	214	140	S ^t Martins Vintrey	417
S ^t Bennet Sherehog	11	1	S ^t Leonard Eastcheap	42	27	S ^t Matthew Fridayth	24
S ^t Botolph Billinggate	83	50	S ^t Leonard Follerslane	335	255	S ^t Maudlins Milkstreet	44
Chrifils Church	653	467	S ^t Magnus Parifi	103	60	S ^t Maudlins Oldfishth	176
S ^t Chriftop hers	60	17	S ^t Margaret Lothbury	100	66	S ^t Michael Baifilthaw	153
Buried in the 97 Parishes within the walls		15207	whereof of the Plague		9887		
S ^t Andrew Holborne	395	3103	Bridewell Precind	20	179	S ^t Dunfons West	958
S ^t Bartholomew Grea	491	344	S ^t Botolph Alderiga	997	755	S ^t George Southwark	1613
S ^t Bartholomew Lette	193	139	S ^t Botolph Algate	4926	4051	S ^t Giles Cripplegate	3069
S ^t Bridget	2111	1427	S ^t Botolph Bilhopfel	454	2500	S ^t Olaves Southwark	4793
Buried in the 16 Parishes without the walls		41351	whereof of the Plague		28888		
S ^t Giles in the Fields	1457	3168	S ^t Katherines Tower	956	601	S ^t Magdalen Bermon	1943
Hackney Parifi	232	132	Lambeth Parifi	798	537	S ^t Mary Newington	1273
S ^t James Claukenwell	863	1377	S ^t Leonard Shorditch	2669	1949	S ^t Mary Iflington	697
Buried in the 12 out-Parishes, in Middlesex and Surrey		8554	whereof of the Plague		21420		
S ^t Clement Danes	1969	13198	S ^t Mary Saoy	303	198	The Total of all the Chriftings	
S ^t Paul Covent Garden	408	261	S ^t Margaret Westminster	4710	3742	9967	
S ^t Martins in the Fields	4804	2883	whereof at the Pefthoufe		1156	The Total of all the Burials this year	
Buried in the 5 Parishes in the City and Liberties of Westminster		12194	whereof of the Plague		8403	97306	
The Difcaies and Casualties this year.							
A Bortive and Stilborne	617	Executed	21	Palfie	30		
Aged	1545	Flox and Small Pox	655	Plague	68596		
Ague and Feaver	5257	Found dead in streets, fields, &c.	20	Plannet	6		
Appoplex and Suddenly	116	French Pox	86	Plurifis	15		
Verid	10	Frighted	23	Poyfoned	1		
Wafled	5	Gout and Sciatica	27	Quinfie	35		
Bleeding	16	Grief	46	Rickets	557		
Bloody Flux, Scowring & Flux	185	Gripping in the Guts	1288	Rifing of the Lights	397		
Burnt and Scalded	8	Hangd & made away themselves	7	Rupture	34		
Calenture	3	Headmould-hot & Mouldfallen	14	Scurvy	105		
Cancer, Gangrene and Fiftula	56	Jaundies	110	Shingles and Swine pox	2		
Canker, and Thrush	111	Impoflume	227	Sores, Ulcers, broken and bruifed	82		
Childbed	625	Kild by feveral accidents	46	Limbs	14		
Chrifomes and Infants	1258	Kings Evil	86	Spleen	1929		
Cold and Cough	68	Leprofie	2	Spotted Feaver and Purples	332		
Collick and Winde	134	Lethargy	14	Stopping of the ftomack	98		
Confumption and Tiffick	4808	Livergrown	26	Surfet	1251		
Convulfion and Mother	2036	Meagrom and Headach	12	Teeth and Worms	2614		
Diftracted	5	Meafles	7	Vomiting	51		
Droplee and Timpany	1478	Murthered and Shot	9	VVenn	2		
Drowned	50	Overlaid & Starved	45				
Christned	Males 5114	Buried	Males 48569	Of the Plague		68596	
	Females 4853		Females 48737				
	In all 9967		In all 97306				
Increased in the Burials in the 130 Parishes and at the Pefthoufe this year				79009			
Increased of the Plague in the 130 Parishes and at the Pefthoufe this year				68596			

Figure 1. An example of a General Bill from 1664-1665 Bills of Mortality. There are parish deaths along the top with specific deaths and casualties along the bottom. In later years, deaths by age category are listed along with the price of foodstuff. Image from the LUNA Folger Digital Collection. Wing (CD-ROM, 1996), G1598A, Goldsmiths', 1761, ESTC (RLIN) R233540

About the Data

The Death By Numbers project has been digitizing and transcribing thousands of London's Bills of Mortality for public use and inquiry. The Bills were originally started in 1603 as a way to track the plague as it spread throughout London. As time progressed, the Bills became more inclusive with the information it displayed. Soon parish statistics, age, types of death or caulties, and foodstuffs were listed on these broadsides. All of this information means that there are multiple sets of data on each broadside that have been separated out. The main focus for the project was tracking the plague and other causes of death in a mathematical sense. This process would create datasets that can be publicly accessed and used by urban historians as well as historians of medicine and other sciences. Information like age and foodstuffs have been interesting to investigate as well.

This project shares its large amount of information to the public through [GitHub](#), a free platform where users from any technical background can use the shapefiles for mapping or data extracted from the Bills themselves. There are also blog posts on our site for any readers to catch up on what we have learned along the way with added visuals when necessary.

Overview of Transformation Process

Metadata Creation, Omeka Import

Transforming the London Bills from archival sources into transcriptions formatted for computational analysis involves a multiple step workflow process. This process includes the digitization of the sources, the import of the images into Omeka S, and the metadata creation. The Bills of London were originally collected and bound into volumes by the [Medical Society of London](#). The Death by Numbers' PI handled the volumes directly and took digital photographs of the Bills. The binding of the London Bills, although relevant for the conservation of the

sources, poses several challenges for the digitization of the documents because the inner margin prevents complete visual access of the bill. The photographs were handled using [Tropy](#), which is a tool designed to manage and describe photographs of research materials.

The metadata needed at this stage of the process included standard fields like name of the source archive, rights information, identifier, and contributor. We created a customized metadata template in Tropy, which also allowed us to enter all images consistently and efficiently in a single batch by using the bulk edit function. It is important to note that the primary function of this metadata was administrative and not descriptive. Its goal was to establish a clear path to backtrack to a specific document if required.

With the metadata created, and the images readily available, we could start the reassembly of the bills. We paired the recto and the verso of each week, and added the metadata elements that are unique, such as the week number and year. The assembly of the bills came with several challenges, including dealing with bills out of chronological order in a given volume, bills bound into the volumes backwards, that is, with the verso before the recto, and illegible text due to obscurity, distorted texts and occasional wormholes.

Finally, we exported from Tropy the customized metadata of all the bills as a CSV file. In the spreadsheet each row corresponds to a bill. The columns contain the specified descriptive data – type of bill, week number, year, source, rights, image filenames, etc. This sheet plays a crucial role in the Omeka S import process. The columns in the spreadsheet were renamed in order to make them more human-readable and to take advantage of the Omeka S ability to automap the metadata. We used the [File Sideload](#) module to import the bills into Omeka S, which required a team member to upload the files to the sideload directory on the same server. Keep in mind that going from Tropy to Omeka S requires adjustments in the file path column in the spreadsheet exported from Tropy. For a more detailed explanation on how to tweak Tropy exports to make it suitable to Omeka S, check the “[From Archival Sources to Computational Analysis. Part Two](#),” blog post on the Death by Numbers website.

We imported the files in Omeka S by organizing them into Omeka S item sets. These item sets were previously designed to match the DataScribe datasets we had in mind to organize the transcription workflow. When we use the Sideload Module, the item sets should be created before the import process starts. We created separated item sets based on the different source materials and ease of metadata collection. We had the Laxton microfilm bills, the Wellcome Library bills, and a set of Bodleian Library bills. This separation was also because each source captures a different characteristic of bills. These item sets ended up populating independent Datasets that are going to be transcribed separately. Once the imports were successfully executed, the items and item sets were ready to be used in the DataScribe model to create as many datasets as the project requires. A successful import process depends on having all the pieces available: a cleaned up and tidy CSV file, the files already located in the server, and the item sets created. Had something gone wrong in the import process, we had all the documentation to rely on, find the mismatch, and run the whole process again.

Dataset Creation with DataScribe

The Bills of Mortality contain a variety of types of data. Each individual bill contains some or all of the following: individual Parish Statistics, Causes of Death city-wide, Gender and Ages of the deceased, and the current price of various Foodstuffs. While all of this information could have been transcribed on the same form, the project team decided these were different enough types of data that they should be treated as separate datasets, which would make it easier to computationally analyze subsets of the data later in the project. We therefore created different datasets with their own forms for each type of data. The item set is the same item set regardless of data entry since our Bills of Mortality Omeka S Items are not separated for the type of data on either side of the sources.

Our dataset Laxton Weekly Bills- Causes is specifically created for a subset of the Bills and to transcribe only the causes of death listed in that item. The side of the Bills listing the Parish Statistics is included in the DataScribe item but is not necessary for transcription. These datasets are all drawn from the same set of bills, and thus were made with the same Omeka Item sets. For example, the Omeka item for the Bill produced during the 5th week of 1708 has individual Parish Statistics, Causes of Death city-wide, Gender data, and the current price of various Foodstuffs; this Omeka item therefore is in a collection that is synced to four different DataScribe datasets, that all exist under the Death By Numbers project umbrella.

When adding a new dataset, the name and itemset are required fields to enter. The name, description, and guidelines are specific to the transcription project for the type of data. Our Foodstuff guidelines are vastly different from our Parish Statistic guidelines since different types of data transcription raise different questions and issues with our transcribers. Some of the guidelines that followed through all of the datasets were reminders of when to use “Is missing” or “Is illegible” checkboxes, reminding transcribers and reviewers to save their work, and reminding all of us that the bills must be transcribed based on the early modern calendar instead of current standards.

Guidelines for the Causes mentioned above were a bit trickier and more specific. Due to the lack of information in some of these cases, we had to decide how much information to infer. When it came to killings, the team decided that accidents should be left under “Other” but all drownings be left together so as to not skew the data for either category. Then the guidelines for Foodstuffs included a basic summary about how bread prices worked in early modern London.

Transcription with DataScribe

Transcription Forms

To create a form with all of the information that will be needed from the document, DataScribe has a form builder section where there is the ability to choose different ways to present the information transcribers must fill in. Depending on the dataset, the forms will look a little different. For the front of the bills with parish information we needed; the parish name, the number of deaths (and plague when necessary), a unique name for the bills including the year and week it was produced, Christenings, and buried.

Unique Identifier *

please indicate the unique identifier for the bill in the format of year-week number, "Laxton-xxxx-xx-verso", e.g. "Laxton-1700-01-verso"

Figure 2. Having the format for transcribers to copy as they work through each form creates consistency and ease of workflow for all levels on the project. The Unique Identifier helped the team keep track of the year and week that specific Bill is from.

The Bills went through several iterations of printing, such as the Laxton Bills versus the Wellcome Bills, and the list of parishes is not fully unique to the type of Bill. Weekly and General bills include all of the parishes which were in the project date range of 1603 to 1752 that had existed at one point due to the fact that some were absorbed or split. That ended up being 149 parishes in the list. For plague years, each parish had a plague and buried count, along with the fields and the subtotals for each section. When we include all of that, it worked out to a form with over 300 fields.

For the bills looking at causes of death that list gets a bit longer. We needed to add all the most common deaths seen for less time spent on form building, the unique name for the bill, then the descriptive section of the bills which included drownings, killed, and more. Some of the

common deaths included cancer, consumption, convulsion, and small-pox but there were so many bills that changed that we felt the full list was easier. Plus, this project is an example of how we used the option to add a checkbox in the forms for missing or illegible information.

Because DataScribe was still in development at the start of the Death By Numbers project, we were able to add a feature to import/export the forms as a JSON file. A JSON file is a text-based format for showing structured data by using JavaScript object syntax. Once that file was exported, DataScribe allowed us to import that back in and change along the way. For a project like the Bills of Mortality, the Import form option is invaluable. In this process alone, DataScribe has saved this project hundreds of hours and ensured the accuracy of the 300 field forms themselves. Being able to import the same form meant that we no longer had to worry about typos or other errors like missing or extra parishes, the always changing causes of death, and descriptive text for the back of the bills.

DataScribe for Project Management

Workflow

DataScribe allows a project team to manage and personalize the workflow as necessary. The Bills of Mortality team had a single graduate student reviewer overseeing a team of undergraduate and graduate transcribers. The reviewer managed the transcription assignments and review process. In this case, the reviewer rarely transcribed after training, but that can vary between reviewers and projects. Other project teams may have multiple people who transcribe and review each other's work. The DataScribe transcription pipeline can be adapted for the workflow and team structure that makes sense for the project.

When assigning work, each transcriber is assigned items to transcribe in batch actions, or items are bulk locked to transcribers. The items are selected with the checkbox next to the

item number and then we can lock them to a specific transcriber. For the Bills of Mortality, we try to lock items by years (according to the item identifier and not the early modern calendar). This workflow happened for a few reasons. The first reason was that if reviewers locked items to specific transcribers, there were not multiple people working on the same Bill and was more time efficient. Second, to lock one whole year to a transcriber ensured that there was a cohesive chunk of data and no gaps in the years of the bills. Lastly, it was helpful in noticing any trends that happened in causes of death. An example of this was that after noticing there were quite a few deaths from falling chimneys in 1703, Katie Kania researched and co-wrote a [blog post on the Great Storm of 1703](#).

Transcribers

Some transcribers adjusted to the workflow faster than others. Some transcribers managed to crank out transcriptions on certain datasets over others. However, all the transcribers will surprise you with unexpected errors or questions about transcriptions since Death By Numbers often deals with messy data, such as descriptive text in addition to numerical entries. We created the guidelines discussed earlier as a living document, but some of the bills had brought up new questions along the way.

Transcribers and reviewers can leave a note at the item level or the record level, and on projects like an account book with many records per item then that distinction may be helpful. But for Death By Numbers, each item should only have one record so transcribers look at the item level since there is not the same obvious distinction between items and records. Our team found it useful to use outside communication like Slack for this as a weekly update system and address immediate problems quickly.

One example of where these workflows and guidelines became necessary for Death By Numbers was the unique way we used the checkbox “Is missing”. Since it was common for parishes to submit their information at a later time, if at all, thus we used the checkbox “Is

missing” when a parish or cause of death was not on a current record as a safe measure. It was also very common to also need the “Is illegible” checkbox for many reasons. Some of the numbers were hard to distinguish due to printing style, ink blots or holes on the paper, and completely illegible due to the microfiche issues. We all had to break the habit of guessing the illegible print due to the goal of the project being accuracy of arithmetic.

Reviewers

The review process of the Bills can change based on the data being reviewed. For both transcription and review, users may find it easier to complete transcription and review by following the Bill’s list of parishes and causes of death or following the DataScribe form’s list. This is somewhat due to the fact that the parish lists on the Bills and the Form were different since the spelling and ordering of parishes changed on different types of Bills. For example, on some Bills, the run of twelve parishes named St. Mary-something were grouped together on the DataScribe form, but were interrupted in a subset of the Bills list by two St. Mary Magdalen-something. On some Bills the Magdalen parishes were listed separately and the DataScribe form was based on that list. Since this difference appeared on multiple years of Bills, or hundreds of transcriptions, the reviewer became familiar with the ordering and decided to scroll the Form and jump around the Bill rather than scroll the Bill and jump around the Form. This is a workflow preference, and transcribers often did the reverse. However, this project proved difficult either way because both the parishes and the causes of death needed to be a full list of all the possible causes and parishes on the Bills for ease of workflow. Out of the roughly 100 causes of death identified by the project and listed on the DataScribe form, each weekly Bill contained anywhere from 20 to 80 of them and were not always in the same order. It would be impossible to make custom forms for each Bill, so the Form needed to contain every possible cause of death. In order to review transcriptions, the

reviewer followed the Bill to ensure the accuracy of each one listed in the transcription record. Again, this is a workflow preference and can be done in the opposite way.

DataScribe for Research

Dataset Export

Once the item is marked as approved, the transcription is added to the CSV export. On some projects, the CSV export can be a secondary review stage to check spelling, formatting, and other transcription issues. However, one of our project goals is to double check the arithmetic on the Bills, therefore incorrect math in the CSV might mean that the transcription is correct. Since the export includes the Omeka item, DataScribe item, and Record number details, any exported transcriptions that appear odd can be checked to identify any problems in the review process.

The first step to export in DataScribe is validating the dataset. This is a process that runs as a background job in DataScribe that checks each of the items to ensure its validity. For example, if a field on an item is supposed to be a number but is instead a string, DataScribe will flag the item for correction. Once the dataset is validated, it can be exported and can be downloaded as a CSV.

Conclusion

We hope that the evidence has shown that Datascribe not only aided in the process of Death By Numbers, but became a crucial part of the process. Being able to take hundreds of Bills from Tropy and batch load them into the Omeka S streamline to Datascribe was one massive help. Then having the ability to create or edit forms quickly and easily to add them to each dataset meant that the transcribers were able to keep working as the PI and project

manager moved on to the next set. Giving instructions in the forms and Guidelines for transcribes to follow allowed for the most accuracy and useability in the data that was extracted from the Bills of Mortality. Finally, having all of that data in an easily downloadable CSV has allowed the team to research and visualize our findings for further study of the Bills of Mortality as well as pre-modern London at large.