

Inequity in Chicago Public Schools (CPS) Supplemental Programs

Technical Report

Balaji Balachandran

Introduction

The work described in the entirety of this document concerns the internship I have held since June 2021, Inequity in Chicago Public Schools (CPS) under the supervision of Dr. Angel Alvarez. This internship identifies disparity in the quality of education that students in the CPS district receive, specifically those attributed to race, socioeconomic status, ability, etc. The analyses I have conducted were all created in Python (3.8), working with large data sets and numerous libraries to create data tables and visualizations that support policy changes proposed to CPS. The link to the source code for the projects can be found on the public GitHub repository linked here: <https://github.com/bbalachandran1/Inequity-In-CPS>. The code for the FOIA request web crawler is denoted with the [FOIA] tag before all relevant files, and the code for the Grade Acceleration Data Analysis is denoted with the [ACCEL] tag. Note: the Grade Acceleration Data Analysis involves large datasets that github does not support. However, the source code that runs the entirety of the analysis, as well as the final analyzed data is on the github link.

Projects

Chicago Public Schools' Publicly Archived FOIA Request Web Crawler

I was first introduced to the Chicago Public Schools' Public Archive of FOIA in August of 2021, as my internship advisor, Dr. Angel Alvarez, complained about the laborious nature of finding the details of individual FOIA requests. The Public Archive of FOIA (Freedom of Information Act) let me and the other individuals on the internship team request data that had already been requested that would aid our analyses. By doing so, it was essentially a guarantee that we could identify the specific FOIA requests that CPS had received and approved, and then request identical data to circumvent reports of requests being "unduly burdensome". I was asked to create a program that would speed up the process of extracting the information from the site, so I set to work on it.

| Request Number | Create Date | Summary | Request Status | Details |
|----------------|---------------------|-------------------------|-----------------------|---------|
| N009830-080421 | 8/4/2021 9:00:00 AM | Security | No Responsive Records | |
| N009827-080321 | 8/3/2021 5:17:00 PM | School Information/Data | Full Release | |

Fig 1. Home page of CPS's Public Archive of FOIA requests

Date Received: 8/3/2021 9:42:54 AM

Name of Requester: Research Development & Compliance Anthony DiRaffaele

Record Description: I am requesting any and all certified payroll records for the project at Lakeview High School 4015 North Ashland Ave. Chicago, IL 60613 General Contractor: Top Builders Inc. Scope of work was VCT installations(Flooring) Project could of happen within the past 2 to 3 years

Status: No Responsive Records

Date Complete: 8/5/2021 1:57:00 PM

Fig 2. Specific details webpage of a given request (N009824-080321 shown above)

I began the project unsure where to start, so I navigated to the website to see how I could possibly extract the details of each request without having to press an arrow that would cause the page to load for a minute at a time (refer to figures 1 and 2).

| A | B | C | D | E |
|----|----------------|-------------|-----------------------------------|----------------------------------|
| 1 | Request Number | Create Date | Summary | Request Status |
| 2 | N009824-080321 | 8/3/2021 | Finance/Budget Records | No Responsive Records |
| 3 | N009823-080321 | 8/3/2021 | Finance/Budget Records | No Responsive Records |
| 4 | N009822-080221 | 8/2/2021 | Other | Partial Denial/Release |
| 5 | N009816-072921 | 7/29/2021 | Employee Information | Full Release |
| 6 | N009808-072821 | 7/28/2021 | Employee Information | Partial Denial/Release |
| 7 | C009805-072721 | 7/27/2021 | Legal Records | Inactive/unknown |
| 8 | N009804-072721 | 7/27/2021 | Facilities/Construction Documents | Full Release |
| 9 | N009803-072721 | 7/27/2021 | Employee Information | Full Release |
| 10 | N009802-072721 | 7/27/2021 | Facilities/Construction Documents | Partial Denial/Release |
| 11 | N009801-072721 | 7/27/2021 | School Information/Data | Portions Released |
| 12 | N009800-072521 | 7/25/2021 | School Information/Data | Partial Denial/Release |
| 13 | N009798-072421 | 7/24/2021 | Employee Information | Full Release |
| 14 | N009792-072321 | 7/23/2021 | Contracts | Exemption Denial |
| 15 | N009789-072121 | 7/21/2021 | Employee Information | Unduly Burdensome-Narrow Request |
| 16 | N009788-072121 | 7/21/2021 | Legal Records | Full Release |
| 17 | N009787-072021 | 7/20/2021 | Other | No Responsive Records |
| 18 | N009786-071921 | 7/19/2021 | Employee Information | Partial Denial/Release |
| 19 | N009785-071921 | 7/19/2021 | Contracts | Exemption Denial |
| 20 | N009784-071921 | 7/19/2021 | Employee Information | Full Release |
| 21 | N009783-071921 | 7/19/2021 | Legal Records | Full Release |
| 22 | N009782-071921 | 7/19/2021 | School Information/Data | Portions Released |
| 23 | N009815-072921 | 7/19/2021 | Legal Records | Partial Denial/Release |
| 24 | N009781-071621 | 7/16/2021 | Instructional Information | No Responsive Records |
| 25 | N009780-071521 | 7/15/2021 | Procurement | Exemption Denial |
| 26 | N009779-071521 | 7/15/2021 | Employee Information | Full Release |
| 27 | N009814-072921 | 7/15/2021 | Legal Records | Partial Denial/Release |
| 28 | N009778-071421 | 7/14/2021 | Other | Withdrawn |
| 29 | N009777-071321 | 7/13/2021 | Procurement | No Responsive Records |

Fig 3. Screenshot of the original spreadsheet downloaded from the CPS Public Archive

I noticed an area that said “Export As XLSX”. The excel sheet that was downloaded (figure 3) contained a general overview of the spreadsheet, but most importantly it contained all of the request numbers, neatly organized into the leftmost column of the sheet. While exploring the website in greater detail, I found a pattern between the request number and the website URL. For request number N009824-080321, the corresponding URL was:



Fig 4. Sample URL of request number N009824-080321; the highlighted portion has a portion of the request number

Characters four through seven, inclusive, of the request number string corresponded to an id that could reference the specific website with the complete details. I created a function using Python and the Pandas library that read the data from the excel sheet “[FOIA]FOIA Requests Empty.xlsx” and inserted the data into lists. The next step was navigating a library I was altogether unfamiliar with: requests. Requests was a library that allowed Python to connect to the internet and extract the HTML source code of any given webpage. I experimented with the library using a sample requests page, isolating the data using my knowledge of HTML tags and separating the information within a <table> tag by using the split string function in Python, effectively separating the source code into smaller chunks I could further isolate to get the useful data. I encountered errors as I first attempted to use the split string function, not fully

understanding the syntax. I believed that `split()` took the source code string and a second string, one that would split the source code string, and split the source code string at the first occurrence of the second string, resulting in two separate strings placed in a list. After much debugging, however, I realized that I fundamentally misunderstood the function, with `split()` actually separating the target string for *every* occurrence of the second string. This meant that I ended up searching for a string that no longer existed when I ran `split()` repeatedly. I fixed my error, now indexing the other rows of the data table of the HTML using the index of the string list. To really isolate all necessary data, I did a similar `split()` process with the ending tag of the HTML table (`</table>`) to truncate the string and produce the final results for the request number. I appended these results to a list, that was then appended to a list of rows to be written to Excel. I ran my program, and much to my amazement there were no errors.

I was optimistic, thinking I could run it overnight and come back with a completed dataset. I returned to my program 5 minutes later, only to see my program had failed after processing seven rows of the Excel file. “`ConnectionResetError: [Errno 104] Connection reset by peer,`” the message read. CPS’s server shut me out (for good reason). I realized I needed a way to circumvent this error while still considering the burden I was placing on their servers, so I tried waiting in between each request. I used Python’s built-in time module to wait 45 seconds between requests, but a quick calculation informed me that it would take ~4 days of my program constantly running to entirely fill out my data sheet. I turned to a better alternative that I had never explored: error handling. I experimented with the socket module to catch the error, at which point the program would wait 60 seconds before trying again. The program now made requests at a quick rate, only stopping when the server protected itself. While creating the error handling functionality, I realized that it was inefficient to begin at the first row every time and progress through the excel file, only to fail a few hundred rows later and not save any progress. To circumvent this oversight, I created a function that asked the user which line to start at and which row to end at, and save once the program had completed all requested rows. The program was now able to complete up to a thousand lines at a time.

However, the program now ran into an Index Error. This error likely occurred because the format of some of the archived websites were irregular. There were only a few rows that experienced this issue, so I simply handled the error similar to how I did for the connection reset error, but also appended the failed row to a list. This list made it so that the information could be manually input into the complete spreadsheet. The program was essentially complete, but as I found myself bored having to wait the sixty seconds when my connection was reset, I got to work creating a fun visual. After all, a good UI is always important. I created an animation from cat ASCII images (figure 5) to occupy the time between handling the connection reset error and before resuming again.



Fig 5. Screenshots of the cat animation pieced together

Once the Index Error was solved, the program was complete, it no longer experienced any errors as it completed the last 3600 rows (I had already completed the data for 4000 rows while testing) with ease, all within 2 hours. The program completed 30 rows per minute, a great improvement over manually filling out the spreadsheet getting one webpage's worth of information in one minute. The resultant spreadsheet is shown below (figure 6). This spreadsheet helped aid research because it helped quickly find and sort data that could be requested, effectively eliminating the work of finding viable data for future analyses. The requests have allowed us to make educated FOIA requests, one the CPS office cannot deny.

| A | B | C | D | E | F | G | H | I | J |
|-------------------|---------------------|---|----------------|-----------------------|--|-------------------------------------|-----------------------|---------------|---|
| Request Number | Create Date | Summary | Request Status | Date Received | Name of Requester | Record Description | Status | Date Complete | |
| 2 N009824-080321 | 2021-08-03 09:42:00 | Finance/Bi No Responsive Records | | 8/3/2021 9:42:54 AM | Research Development & Compliance Anthony Di I am requesting any a No Respor | 8/5/2021 1:57:00 PM | | | |
| 3 N009823-080321 | 2021-08-02 09:39:00 | Finance/Bi No Responsive Records | | 8/3/2021 9:39:52 AM | Research Development & Compliance Anthony Di I am requesting any a No Respor | 8/5/2021 1:55:00 PM | | | |
| 4 N009822-080221 | 2021-08-02 15:26:00 | Other Partial Denial/Release | | 8/2/2021 3:26:25 PM | Gerardo M. Mendez | To Whom it May Con Partial Der | 8/9/2021 3:17:00 PM | | |
| 5 N009816-072921 | 2021-07-29 11:38:00 | Employee Full Release | | 7/29/2021 11:38:36 AM | L. FitzPatrick | Under the Illinois FOI Full Releas | 8/6/2021 1:22:00 PM | | |
| 6 N009808-072821 | 2021-07-28 10:03:00 | Employee Partial Denial/Release | | 7/28/2021 10:03:43 AM | Johnna M Richmond | I am doing a backgroun Partial Der | 8/4/2021 4:00:00 PM | | |
| 7 C009805-072721 | 2021-07-27 19:45:00 | Legal Recr Inactive/unknown | | 7/27/2021 7:45:43 PM | Ghansham Patel | I would like records c Inactive/u | 7/28/2021 8:37:00 AM | | |
| 8 N009804-072721 | 2021-07-27 19:28:00 | Facilities/Full Release | | 7/27/2021 7:28:08 PM | Dr. John Kugler | supply the following i Full Releas | 8/6/2021 2:29:00 PM | | |
| 9 N009803-072721 | 2021-07-27 17:08:00 | Employee Full Release | | 7/27/2021 5:08:20 PM | Dr. John Kugler | We request a copy or Full Releas | 8/6/2021 4:52:00 PM | | |
| 10 N009802-072721 | 2021-07-27 17:00:00 | Facilities/Partial Denial/Release | | 7/27/2021 5:00:10 PM | Dr. John Kugler | We request a copy or Partial Der | 8/6/2021 2:25:00 PM | | |
| 11 N009801-072721 | 2021-07-27 13:51:00 | School Inf Portions Released | | 7/27/2021 1:51:46 PM | Dr Carolina Barrera Tobón | I am looking for infor Portions R | 8/4/2021 12:41:00 PM | | |
| 12 N009800-072521 | 2021-07-25 15:42:00 | School Inf Partial Denial/Release | | | | | | | |
| 13 N009798-072421 | 2021-07-24 16:06:00 | Employee Full Release | | 7/24/2021 4:06:29 PM | Lara Stachler | I would like the 2021 Full Releas | 8/10/2021 12:40:00 PM | | |
| 14 N009792-072321 | 2021-07-23 10:03:00 | Contracts Exemption Denial | | 7/23/2021 10:03:48 AM | Rachel Jacobs | Id like to request a di Exemption | 7/27/2021 10:35:00 AM | | |
| 15 N009789-072121 | 2021-07-21 09:05:00 | Employee Unduly Burdensome-Narrow Request | | 7/21/2021 9:05:39 AM | Human Resources Administrative Assistant Nicholl | I am requesting a phc Unduly Bu | 7/26/2021 2:42:00 PM | | |
| 16 N009788-072121 | 2021-07-21 06:41:00 | Legal Recr Full Release | | 7/21/2021 6:41:30 AM | Carl A Willecke | Ms. Diaz, In the spirit Full Releas | 7/27/2021 6:07:00 PM | | |
| 17 N009787-072021 | 2021-07-20 15:49:00 | Other No Responsive Records | | 7/20/2021 3:49:04 PM | Organizer Ryan Hollander | 1. Would you please No Respor | 7/23/2021 4:30:00 PM | | |
| 18 N009786-071921 | 2021-07-19 19:43:00 | Employee Partial Denial/Release | | 7/19/2021 7:43:23 PM | | Please supply the full Partial Der | 8/3/2021 4:37:00 PM | | |
| 19 N009785-071921 | 2021-07-19 16:00:00 | Contracts Exemption Denial | | 7/19/2021 4:00:42 PM | | NWEA respectfully re Exemption | 7/27/2021 10:33:00 AM | | |
| 20 N009784-071921 | 2021-07-19 15:53:00 | Employee Full Release | | 7/19/2021 3:53:01 PM | Steven Britton | I would like the name Full Releas | 7/29/2021 11:40:00 AM | | |
| 21 N009783-071921 | 2021-07-19 10:49:00 | Legal Recr Full Release | | 7/19/2021 10:49:40 AM | Miriam Bhimani | Since November 1, 20 Full Releas | 8/2/2021 1:12:00 PM | | |
| 22 N009782-071921 | 2021-07-19 10:14:00 | School Inf Portions Released | | 7/19/2021 10:14:06 AM | IYS Specialist Marni Basic | List of all CPS schools Portions R | 7/28/2021 10:52:00 AM | | |
| 23 N009781-071921 | 2021-07-19 09:00:00 | Legal Recr Partial Denial/Release | | 7/19/2021 9:00:00 AM | Cleother Tidwell | 1)"Jamari Dent suffe Partial Der | 7/29/2021 3:37:00 PM | | |
| 24 N009781-071621 | 2021-07-16 10:52:00 | Instructor No Responsive Records | | 7/16/2021 10:52:31 AM | | I would like instructic No Respor | 7/23/2021 2:57:00 PM | | |
| 25 N009780-071521 | 2021-07-15 14:58:00 | Procurement Exemption Denial | | 7/15/2021 2:58:25 PM | Rachel Jacobs | I would like to please Exemption | 7/27/2021 10:31:00 AM | | |
| 26 N009779-071521 | 2021-07-15 13:51:00 | Employee Full Release | | 7/15/2021 1:51:01 PM | L. FitzPatrick | A work history for th Full Releas | 7/29/2021 2:55:00 PM | | |
| 27 N009814-072921 | 2021-07-15 09:00:00 | Legal Recr Partial Denial/Release | | 7/15/2021 9:00:00 AM | Cleother Tidwell | Requesting the lawsu Partial Der | 7/29/2021 3:30:00 PM | | |

Fig 6. Completed spreadsheet with FOIA request descriptions and all details

Grade Acceleration Data Analysis

The goal of the grade acceleration data analysis was to identify inequity in the way students of the Chicago Public School district were allowed to skip grades. Currently, students in CPS were required to score higher than the 95th percentile nationally to be allowed to skip a grade. However, identifying how students performed in comparison to students at their own school places students at a more equal playing field, and thus should be the measure for grade acceleration. It was this thinking that drove my Grade Acceleration Data Analysis. I looked at NWEA test score data from 2017-2020, a standardized test given to students across the nation to evaluate their learning. It was a large dataset, containing over five million rows for each year, and four years to work with, one that even pandas struggled to process quickly.

First to separate data points by school, I created a function that would find the list of all unique schools using an ordered set of the data entries in the school column and then reverting them into a list again. This gave me a list of unique schools I compared to each individual row, which allowed me to append each row into a list. By doing this, I could now specifically filter the testing data by school. My next step was to perform a similar filter based on grade level and subject, using a more simple process since the amount of options for grade level and subject were limited. But this was easier said than done. I struggled to implement for loops as my mind turned over remembering which list was which, the order getting constantly lost. Eventually I sorted the data into the necessary categories, and could begin running an analysis. The data now took the form of a six-dimensional array, one that I learned to navigate using specific indices.

Next, I accumulated the test scores for each school, grade level, and subject using a series of for loops. I calculated important statistics for the grade acceleration of a student with a series of if statements, comparing each individual student's test score to the average score of students two grades above, the 90th percentile score of students one grade above, etc. I collected the national 95th percentile for each subject and grade from the NWEA 2015 report, using the pandas library as a means of reading the data from the sheet. Once counting and comparing all of the data with the series of for's and if's, I inserted my analysis in rows: one for each combination of subject, grade-level, and school. I ran the program, expecting beautiful results, only to be met with a sea of "N/A". Two dimensional lists gave me pause at times, but five? Those had me struggling on a single line of code for hours. As I completed my first

analysis, I realized a tremendous oversight, I never accounted for differences in test season. Counting the scores that exceeded a certain percentile meant nothing if students tested in the spring were compared to students tested in fall. I dreaded myself for a moment, not eager to make a *seven* dimensional array, when I realized: I could just filter it out with Excel. I put aside my programming pride as I manually separated the data into fall, winter, and spring. Finally, I ran the program with reasonable results. The analysis spanned nearly ten thousand rows, but that was measly in comparison to the twenty million at the beginning (see figure 7).

| School | Grade | Subject | # of Students in the Grade | Mean Score of Students in the Grade | 90th Percentile Score of Grade Above | 50th Percentile Score of 2 Grades Above | 90th Percentile Score of 2 Grades Above | # of Students Scoring Above 90% 1 Grade Higher | # of Students Scoring Above 50% 2 Grades Higher | # of Students Scoring above 90% 2 Grades Higher | 95th Percentile Score Nationally | # of Students Scoring above 95% Nationally |
|----------|-------|-------------|----------------------------|-------------------------------------|--------------------------------------|---|---|--|---|---|----------------------------------|--|
| BRADWELL | K | Mathematics | 40 | 128 | 169 | 161 | 179 | 0 | 0 | 0 | 165 | 0 |
| BRADWELL | K | Reading | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | 163 | N/A |
| BRADWELL | 1 | Mathematics | 51 | 150 | 179 | 178 | 193 | 1 | 1 | 0 | 184 | 1 |
| BRADWELL | 1 | Reading | N/A | N/A | 171 | 170 | 196 | 0 | 0 | 0 | 182 | N/A |
| BRADWELL | 2 | Mathematics | 12 | 161 | 193 | 202 | 213 | 0 | 0 | 0 | 199 | 0 |
| BRADWELL | 2 | Reading | 11 | 160 | 196 | 200 | 214 | 0 | 0 | 0 | 200 | 0 |
| BRADWELL | 3 | Mathematics | 5 | 178 | 213 | 203 | 210 | 0 | 0 | 0 | 212 | 0 |
| BRADWELL | 3 | Reading | 4 | 170 | 214 | 204 | 217 | 0 | 0 | 0 | 214 | 0 |
| BRADWELL | 4 | Mathematics | 13 | 202 | 210 | 207 | 221 | 3 | 3 | 0 | 225 | 0 |
| BRADWELL | 4 | Reading | 13 | 200 | 217 | 203 | 221 | 0 | 5 | 0 | 224 | 0 |
| BRADWELL | 5 | Mathematics | 7 | 203 | 221 | 217 | 227 | 0 | 0 | 0 | 236 | 0 |
| BRADWELL | 5 | Reading | 7 | 204 | 221 | 210 | 220 | 0 | 2 | 0 | 231 | 0 |
| BRADWELL | 6 | Mathematics | 51 | 207 | 227 | 219 | 237 | 1 | 9 | 0 | 243 | 0 |
| BRADWELL | 6 | Reading | 53 | 203 | 220 | 221 | 231 | 7 | 5 | 1 | 236 | 0 |
| BRADWELL | 7 | Mathematics | 3 | 217 | 237 | N/A | N/A | 0 | N/A | N/A | 250 | 0 |
| BRADWELL | 7 | Reading | 3 | 210 | 231 | N/A | N/A | 0 | N/A | N/A | 240 | 0 |
| BRADWELL | 8 | Mathematics | 3 | 219 | N/A | N/A | N/A | N/A | N/A | N/A | 256 | 0 |
| BRADWELL | 8 | Reading | 3 | 221 | N/A | N/A | N/A | N/A | N/A | N/A | 243 | 0 |

Fig 7. Sample table of data that finds the number of students that would meet the qualification given the filters

But the real magic came when I combined my affinity with Python and a niche understanding of Excel: the 3D map. I plotted the data and the statistics spoke for itself.

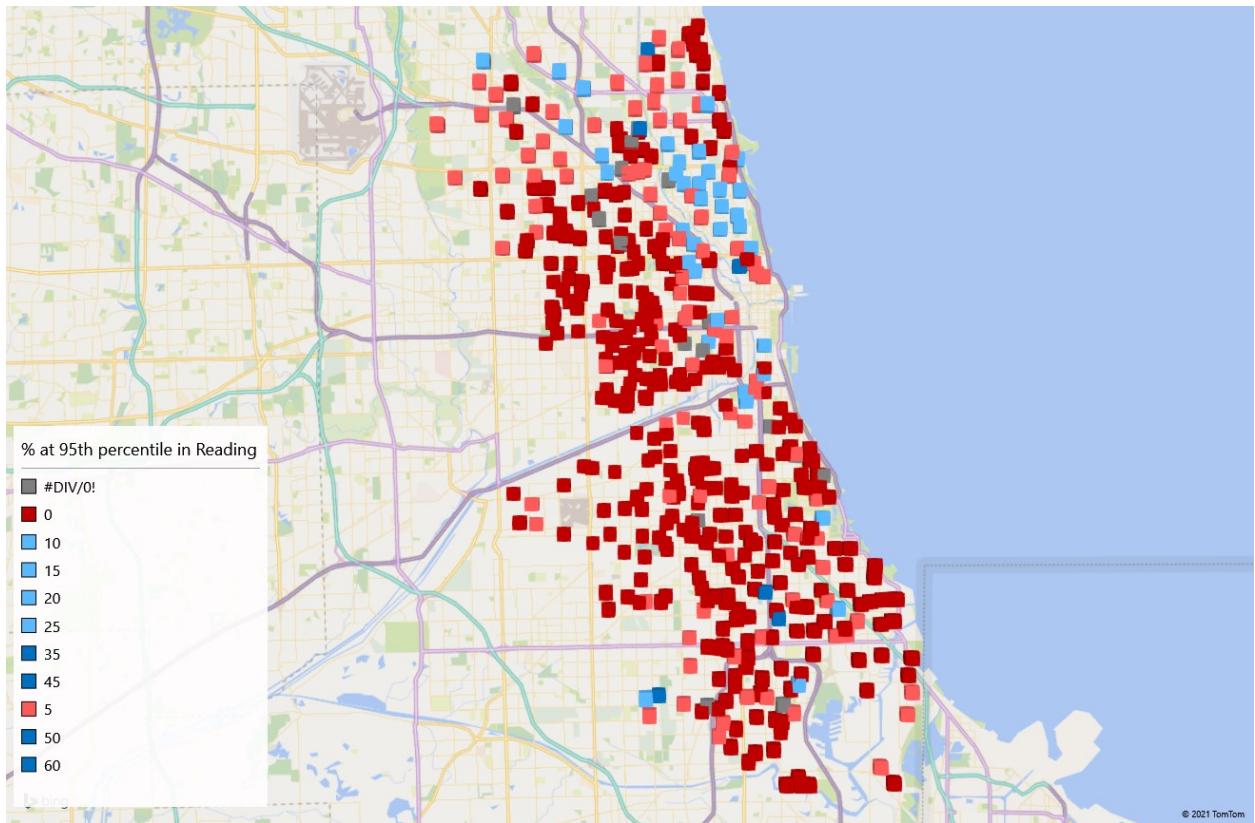


Fig 8. Plotting the percentage of students exceeding the 95th percentile of the nation in reading

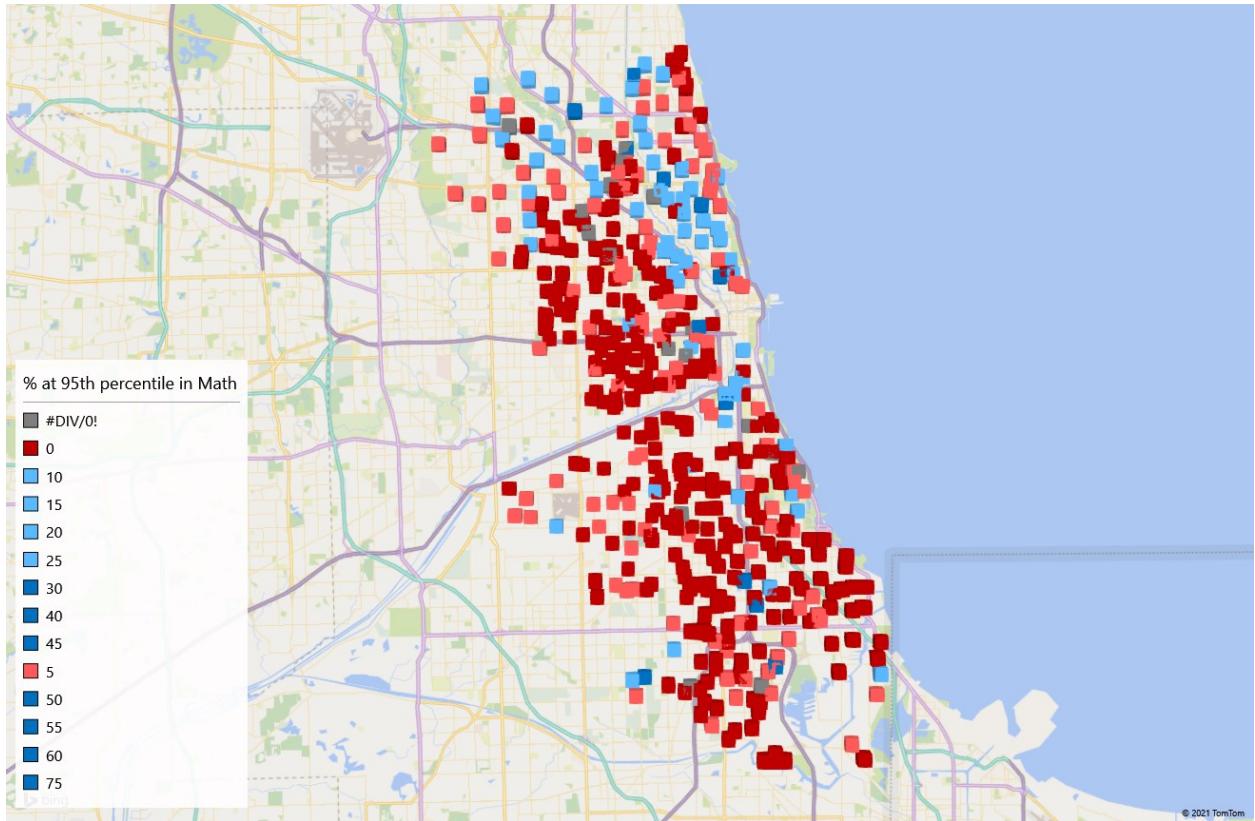


Fig 9. Plotting the percentage of students exceeding the 95th percentile of the nation in math

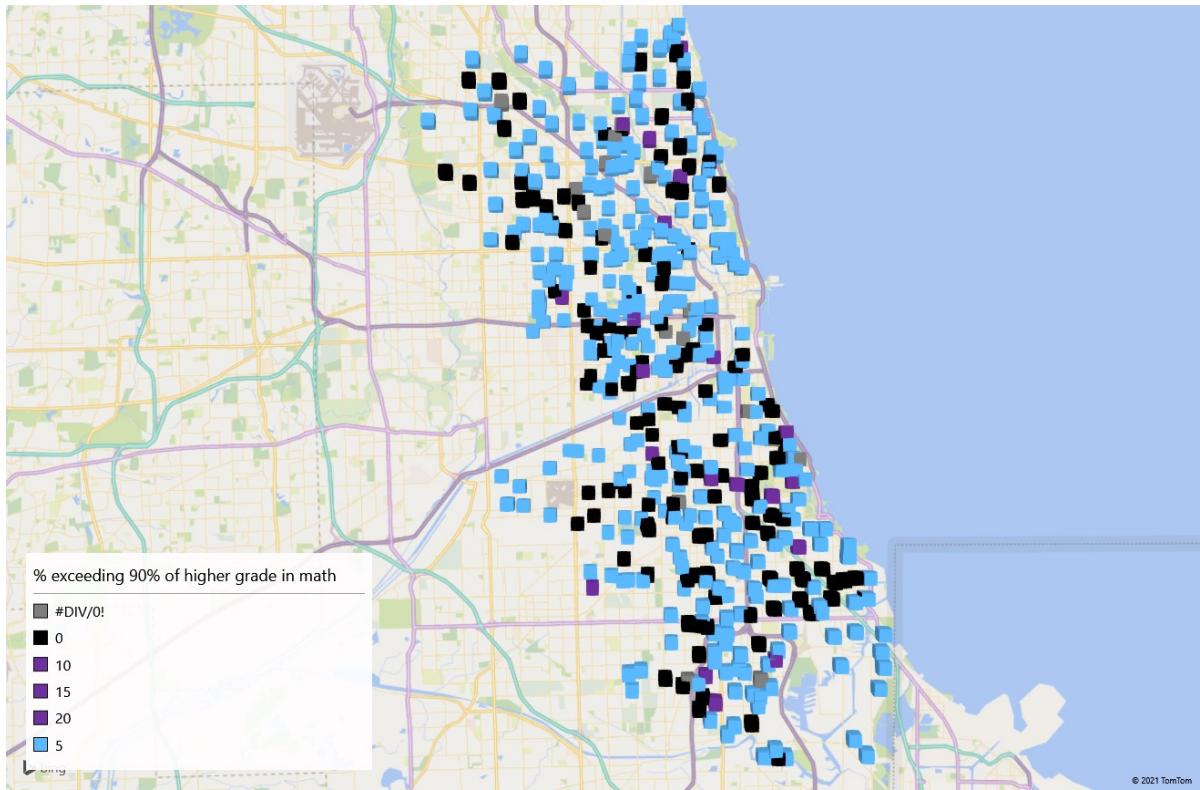


Fig 10. Plotting the percentage of students exceeding the 90th percentile of the grade above *at their school*

While not related to the technical nuance of the project, I'd like to end on a final note about the final graphs (figures 8-10) and their relevance. Students in CPS were required to score higher than the 95th percentile nationally to be allowed to skip a grade. The analysis showed that few students on the South and West sides could accelerate a grade, contrasted by a sizable minority of students on the North side, represented by a light blue streak. However, the data was misleading, implying students on the south and west side would not qualify to skip a grade. However, a comparison to the 90th percentile of students within their *own* school told a different story (figure 10). Students from all parts of Chicago were exceeding the 90th percentile of students in the grade above in a similar proportion, a constant pattern. From 100 million entries of data came a brilliant visual: it was inequity holding these underserved students back.