A Snapshot of Computer Science in K-12 Public Schools in New York City

Katherine Rojas and Maeghan Sill

Learning Analytics, Department of Human Development, Teachers College Columbia University, kr2678@tc.columbia.edu

Learning Analytics, Department of Human Development, Teachers College Columbia University, mms2380@tc.columbia.edu

Abstract

Computer science education is a quickly expanding field. Students studying computer science education in primary and secondary school have better outcomes in core classes and soft skills assessments. These students are also more likely to get a bachelor's degree in a STEM field, giving them increased opportunities for high-paying jobs. In 2015 New York City passed CS4All, an initiative that committed to providing computer science education for all students from grades K-12. This research article seeks to identify trends in computer science education in New York City public schools in the seven years since then.

This study examines how the demographics of students taking computer science classes have changed over time, revealing a steady increase in enrollment across all grade groups, English Language Learners, and economically disadvantaged students after 2020-2021. Next, the study took an analytical look at the number of computer science courses offered per school at different grade levels. The data reporting in this area is incomplete, and its impact on our findings is discussed in full throughout the results section. Next, the study found no relationship between a school's bandwidth capacity and the number of computer science courses offered at the school, suggesting factors beyond technological infrastructure influence computer science course offerings. Finally, the research explored the influence of STEM teachers on computer science course availability. In high schools, STEM teachers and computer science course offerings correlate, implying that STEM educators may have some influence on computer science course offerings at a school.

Keywords

computer science education, demographic trends, course offerings, educational equity, STEM teachers

1. Introduction

As technology advances around us at an unparalleled pace, helping students be best prepared for the challenges of tomorrow can seem daunting and alluding. What tools and skills will help the next generation solve the complex problems they will face? Many argue that the answer is computer science classes. Though it is often confused with digital literacy, computational thinking, computer programming, and algorithmic thinking, Florez et al. say that computer science is a field that "incorporates techniques and methods for solving problems and advancing knowledge (such as abstraction and logical reasoning), and a distinct way of thinking and working that sets it apart from other disciplines" (2017). They explain that computational thinking, the type of problem-solving taught when learning computer science, applies to all areas of life. Both of these are different from computer programming, which is the act of communicating instructions to a machine. The New York City Department of Education provides an alternative definition: computer science is "the study of the capabilities and limitations of computers" (2022). They further emphasize that learning computer science helps develop computational thinking. Another definition, provided by Vegas and Fowler asserts that it is "the study of both computer hardware and software design including theoretical algorithms, artificial intelligence, and programming (2020)."

According to the U.S. Bureau of Labor Statistics, computer and information technology (CIT) will be one of the fastest-growing fields over the next ten years. (Oldham, 2020). Currently, the median salary for jobs in this field is twice the median U.S. salary (Oldham, 2020). Additionally, even careers outside of the CIT field are beginning to require more knowledge of computer science (Slagg, 2022). Between 2002 and 2016, the number of jobs in the United States that require medium-high digital skill levels rose from 44 percent to 70 percent (Ezell, 2021). As students entering the workforce with computer science skills will have more opportunities, it only makes sense that stakeholders of the U.S. education system, such as parents, teachers, students, principals, policymakers, and educational reform advocates, have been pushing toward more computer science education for K-12 students. The rest of the world has also been pushing to expand computer science access in primary and secondary schools. However, in 2020, Vegas and his colleagues did online research on 219 countries worldwide. They found that only 20 percent had policies mandating each school to have a computer science course as either an elective or a required course (2020). In the United States, as of November 2023, only seven states required a course in computer science to satisfy high school graduation requirements (Stiffler, 2023).

Amidst this current push, some still question or disagree with promoting computer science for all students. Larry Cuban, a professor emeritus of education at Stanford University, argues, "Why would you teach coding to little kids, or even big kids, unless they want to be programmers? (Oldham, 2021)" He advocates for teaching it like a vocation alongside woodworking, shop, and drafting classes. Still, supporters of policies to advance computer science access in K-12 education cite more than just future opportunities as motivation. There is substantial evidence that studying computer science in childhood and adolescence improves learning outcomes and soft skill development. For example, students who take programming courses perform better on assessments of logical reasoning and problem-solving skills (Florez et al., 2017). Computational thinking courses have been shown to lead to better response inhibition and planning abilities (Vegas et al., 2020). A report co-created by the Code.org Advocacy Coalition, the Computer Science Teacher Association, and the Expanding Computing Education Pathways Alliance stated that there is a correlation between learning computer science skills and better outcomes in math, science, and reading classes (cited in Arundel, 2023). This may be due to the explicit teaching of computational thinking and problem-solving skills in computer science classrooms.

Many proponents also see computer science as a way for the United States to stay competitive in the global economy. Once a leading developer of STEM talent, the U.S. is forecasted to produce half the number of STEM doctorates as China by 2025 (Howell, 2023). Out of the 219 countries researched by Vegas's team in 2020, the U.S. was one of the 7 percent, about 15 countries, that offered computer science in some schools and some jurisdictions to varying degrees. The U.S. has gaps in STEM knowledge compared to our international counterparts. 15-year-olds who took the PISA in 2018 scored 18th in science literacy and 27th in math literacy out of 77 (Howell, 2023). The results of the 2019 Trends in International Mathematics and Science Study (TIMMS), an international assessment given every four years, showed that fourth graders in America ranked 15th out of 64 in math and 8th out of 64 in science (U.S. Department of Education). U.S. 8th graders ranked 11th out of 46 in math and 8th out of 46 in science (U.S. Department of Education). Among many other politicians and analysts, Howell posits that developing a workforce capable of creating and managing technological advancement ethically is a matter of national security (2023).

These reasons inform public sentiment, which is, across the board, in support of students of all ages having access to computer science education. According to a report by Google, Inc. and Gallup, Inc. 2016, ninety percent of parents believe that providing computer science educational opportunities is a good use of school resources. Teachers, principals, and superintendents also agree with this statement at 71 percent, 66 percent, and 65 percent, respectively. Sixty percent of educators also agreed with requiring most students to take a course in computer science if it is offered at their school. Students even show interest in computer science, with 94 percent agreeing that learning about computer science allows you to "work on fun and exciting projects" and 85 percent of students thinking it is very or somewhat likely that they will need to know computer science for their future job.

With a majority of Americans behind the movement, in 2016, President Obama began the initiative Computer Science for All. This plan called for more than four billion dollars to be directed toward expanding K-12 computer science education with the goal of:

> All American students from kindergarten through high school to learn computer science and be equipped with the computational thinking skills they need to be creators in the digital economy, not just consumers, and to be active citizens in our technology-driven world. (Smith, 2016)

New York City launched an initiative of the same name one year earlier, in 2015 (Dvorkin & Crowe, 2023). On the program's website, The New York City Department of Education states the goal of the initiative:

> Computer Science for All will ensure that all NYC public school students learn computer science, with an emphasis on students who identify as girls, Black and Latinx students. Through our work, students will be better prepared to utilize computer science during their K-12 experience and after graduation. (2022)

The emphasis that New York City has placed on reaching marginalized students is necessary. Currently, only 31% of students in high schools nationwide are women (Code, 2023, as cited in Stiffler, 2023). The National Science Board has found gaps across socioeconomic status, race, ethnicity, and sex (Howell, 2023). They have also found that schools with low socioeconomic status have less highly qualified teachers. A study by Change the Equation found that students attending high-poverty schools have a lower chance of encountering STEM throughout their education (Howell, 2023). This mixture of lack of high-quality opportunities eventually leads to marginalized students graduating from college with STEM degrees at much lower rates than other groups (Howell, 2023).

Understanding the importance of computer science and high-quality opportunities for STEM exposure, we decided upon four research questions to understand how New York City is implementing its initiative:

1. How have the demographics of students taking computer science changed over time?
2. Do computer science course offerings differ by borough?
3. What is the relationship between school bandwidth and computer science course offerings?
4. Does the number of stem teachers in a school in a given year correlate to the number of computer science classes offered?

2. Research Methodology

Our research used data shared on the NYCED Information Hub and NYC Open Data websites to analyze computer science education in New York over a seven year period. "Local Law 177 enacted in 2016 requires the Department of Education of the New York City School

District to submit to the Council an annual report concerning computer science education" (*Computer Science and Career and Technical Education Reports*, n.d.). The reports for each year provided a plethora of information about the number of computer science courses offered by school (Sheet 1), number of students enrolled in computer science programs (Sheet 2), the number and ratio of certified STEM instructors (Sheet 3) and the total available bandwidth in each school (Sheet 6). Sheets 4 & 5 included information on STEM Institutes and Computer Science Initiatives. There wasn't much usable information in these tabs and only 2-3 years had any information in them. The data file for each year contained the aforementioned data separated into individual sheets. Our team downloaded the data for each year and separated the sheets into individual files to be merged, organized and analyzed using Google Collab.

2.1. & 2.2 Data Source & Sample

Our study used several data sets merged together to give us a picture of computer science education in New York. Below is a breakdown of each data sample organized by research question.

RQ1 - How have the demographics of students taking computer science changed over time?

For this question we used Sheet 2 data (number of students enrolled in computer science programs). We merged each sheet from 2015 - 2022 (grouped by year) providing us with seven years worth of data. Sheet 2 provided us with counts for # of Students, # Students Taking CC, % Students Taking CS, and % Within CS by grade band (all grades, grades K-5, grades K-8 and grades 9-12). The rows for the data of these grade bans were divided up into categories such as Community School District (33 in total), Poverty Status (Economically disadvantaged or not), English Language Learner Status (ELL vs Non-ELL), Ethnicity(Group Identification), Gender (male or female) and Student with Disability Status (with or with a disability). This gave us 48 rows and 19 columns of information for each year. The completely merged dataframe grouped by year contained 356 rows worth of information. Each year contained summary information for each category for over 900,000 students (average of 959,246 students). It is important to note that reports are required but there is still missing data that was labeled as 's' on various sections as well as 'Nan'. The data types were all objects which eventually needed to be changed to integers or floats for analysis. The data consisted of categorical variables and both counts and percentages by grade band and participation in CS courses. For this research question we primarily focused on using the percentage data as it gave us a holistic view of the percentage of students taking CS by categorical variable.

RQ2 - Do computer science course offerings differ by borough?

For this inquiry, Sheet 1 data (the number of computer science courses offered by school) was referenced. We merged each sheet from 2015 - 2022 (grouped by year) providing us with seven years worth of data for this question as well. Sheet 1 provided us with counts for # of Comp Sci Courses, # of AP Comp Sci Courses, # of Full CS Courses, and # of partial CS Courses by District Borough Number (DBN). DBN is the combination of the District Number, the letter code for the borough, and the number of the school. The rows for the data were each school that reported the number of courses by type for their school. This gave us an average of 526 rows for each year (15-16 having the least rows at 236 and 21-22 having the most at 710) and 6 columns of information for each year. The completely merged dataframe grouped by year contained 3749 rows worth of information. We ended up only using the DBN, # of Comp Sci and AP course columns from the original source. Our team had to create the Year column to help group the data as well as a Borough column to add another categorical variable. The school name column was deleted since the DBN number was enough of an identifier. It is important to note that reports are required but there is still missing data. The data types were all objects which eventually needed to be changed to integers or floats for analysis. The data consisted of categorical variables and both only counts of number of courses per school. For this research question we primarily focused on using 2 course types as full and partial CS courses would still count as a computer science course.

RQ3 - What is the relationship between school bandwidth and computer science course offerings?

This research question used the data from RQ2 in addition to data from Sheet 6 (bandwidth data for each school) from 2015 - 2022 (grouped by dbn). Sheet 6 provided us with dbn, building code, school name and counts for bandwidth. We deleted the building code and school name columns as the dbn is a unique value. After merging the bandwidth for each year using the dbn's as the common variable this gave us 2595 rows and 8 columns of information. The completely merged by dbn dataframe of school bandwidth and computer science course offerings grouped by year contained 10996 rows worth of information. It is important to note that reports are required but there is still missing data that was labeled 'Nan' as the number of schools in 15-16 was 236 and in 21-22 was 710. The data types were all objects which eventually needed to be changed to integers or floats for analysis. The data consisted of categorical variables and counts by dbn and participation in CS courses.

RQ4 - Does the number of stem teachers in a school in a given year correlate to the number of computer science classes offered?

For this section we used the data from Sheet 3 (the number and ratio of certified STEM instructors) for the same year period (2015-2022). Sheet 3 contained information on the number of students for each school that reported, school type, # of full time stem teacher, and # part-time stem teacher. We removed the school name and ratio columns. We merge the stem teacher data into one larger data frame grouped by year. The resulting data frame had 11254 rows of data stored as objects, integer and float values. In order to see the relationship between stem teachers and computer science course offering we eventually merged the computer science course offering by dbn number. This final dataframe has 12447 rows of information and 16 columns and it mostly consists of float data, 1 int and 1 object.

2.3. EDM Methods

In order to analyze the four final merged datasets for our four research questions our team had to use a combination of descriptive statistics, regression analysis (OLS), and correlation mining to find the relationships between computer science and student demographic data over time, differences in course offerings by borough, relationship between school bandwidth and CS course offerings and correlation between the number of STEM Teachers and CS classes offered.

RQ1 - Demographic Changes Over Time

Descriptive Statistics: We use this method to summarize the demographic data of students taking computer science (CS) courses each year. We focused on the calculated mean of the percentage of students taking CS by specific demographic category over time. By using descriptive statistics we were able to find trends in the mean % of students taking CS by year and color coordinating the graph to be not just grade specific but demographic category specific. The other demographics we were able to analyze were District, Borough, English Language Learners, Economic Status, Ethnicity, Gender and Disability Status. We used bar charts and demographic grouped line plots to show the change over time.

Regression Analysis (OLS) & Correlation: OLS was used to analyze the change in the % of students taking CS across different dimensions (school groups, district, borough, English Language Learners (ELL), Socioeconomic Status (SES), ethnicity, gender, disability) over time. We perform OLS regression to quantify changes over time by treating time (year) as the independent variable and the percentage of students taking CS as the dependent variable. We integrated through each demographic/categorical value and identified the demographic level with the largest coefficient indicating most significant change over time. We also assessed p-values for statistical significance.

RQ2 - Relationship between Course Offerings by Borough

Descriptive Statistics: By summarizing the number of CS course offerings by borough, descriptive statistics provided insight into how course availability varies geographically. We used a bar plot to show which borough had the highest average number of computer science courses.

Regression Analysis (OLS) & Correlation: OLS allowed us to estimate the quantitative effect of being in one borough versus another on the number of course offerings. In this case the borough was the categorical feature in the model. The independent variable was the year and the dependent variable the number of Computer Sci Courses. The Coefficient and P-Value allowed us to assess the strength and statistical significance of the relationship.

RQ3 - Relationship Between School Bandwidth and CS Course Offerings

Descriptive Statistics: We used .describe to explore the bandwidth data over the years and the CS course offerings data separately first to see if there were any trends. We looked at the mean, and distribution of both bandwidth and CS course offerings across schools. We also visualized the results using a scatter plot and histogram.

Regression Analysis (Seaborn Inplot - Linear Regression Model) & Correlation: This regression analysis was helpful in visually showing how the relationship between bandwidth availability and the number of computer science courses offered has evolved over time, with each year's data visualized in its own subplot for comparison. We also computed a correlation coefficient by year to describe the strength and direction of the relationship between bandwidth and CS offerings.

RQ4 - Correlation Between Number of STEM Teachers and CS Classes Offered

Descriptive Statistics: We used this to visualize and summarize the number of STEM teachers and CS classes across the schools we paid close attention to count and mean.

*Regression Analysis (Seaborn Inplot - Linear Regression Model) & Correlation: T*his regression analysis was also helpful for this question in visually showing how the relationship between stem teachers and the number of CS courses. We also took the opportunity to not just visualize the relation by year but also by school type (elementary, secondary, etc).

2.4. Data Analysis

RQ1 - How have the demographics of students taking computer science changed over time?

Since each research question worked with a different dataset and each data set had csv's for multiple years, our first step was to import the data and see what cleaning was necessary. We cleaned each year's data set one by one to make sure the column headers and lined up for merging later. We went through the process of deleting any extra columns that were added but were blank. Next we renamed the data columns as they all shifted to rows 4 and 5 of the data frame. After we added a new column to each data frame to identify the data being imported for that year. We proceeded to merge the demographic data for each year into one master data set using pd.concat. Now that we were not dealing with 7 separate data sets, we replaced any non reported data that has the value 's' with 'Nan', removed any commas from numbers, as well as removing any columns we did not need for the analysis. Our last step in using the master data frame was checking the data type and making the necessary conversion to floats and integers for our descriptive statistics and regression analysis to run properly.

Since our research question takes into consideration demographic data while assessing students taking computer science classes over time, we separated our main data set into smaller datasets grouped by their demographic/category type (Community School District, Poverty Status, English Language Learner Status, Ethnicity, Gender and Student with Disability Status). This allowed us the opportunity to run descriptive statistics to get the mean % of students taking CS for that demographic. We used the .groupby function to regroup the demographic data by year with relation time. This allows us to create several bar plots visualizing the mean % of students taking CS by Demographic Value over time.

It is important to note that we had to convert our year range 2015-2016 to just 2015 in order to run the regression. In order to run the regression model we have to iterate through each specific demographic and then perform the analysis. Our X variable was year and our y was the % of students taking CS across all grades. We used OLS as the descriptive statistics alluded to a linear relationship. Output included Demographic Value (district, borough, gender etc) with the most significant change over the time, coefficient for each demographic value and p-values as well. Pyplot from matplotlib was used to graph group line plots to visualize our findings.

RQ2 - Do computer science course offerings differ by borough?

For this research question we went through similar steps as RQ1 to organize and clean the data. We used python functions to drop columns and rename the column headers for merging the data sets from 2015-2022. After the format for each data set was the same we added a year column to each to help distinguish which dataset was from which year. We then used pd.concat again to merge the data sets and stack them vertically. We then checked data types to convert to int and floats for analysis. Since the DBN numbers in this data set contain letters indicating the NYC borough the school is in, we converted the dbn column to a string to extract the letter corresponding with the borough and output the full borough name in a new column. Now that our dataset is ready we looked at descriptive statistics specifically value counts and the means. We extracted the count column to create a bar plot with the average number of computer science courses by borough from 2015 - 2022.

Afterwards we performed our regression analysis. For this we grouped the data frame by borough and year. We then created a group line plot where we integrated over each borough and plotted the line for not just # of computer science courses but # of AP Computer Science courses as well. We visualized the results by plotting the mean # of CS courses by Borough and Year. Each line on the graph represents a borough and class type (regular or AP).

RQ3 - What is the relationship between school bandwidth and computer science course offerings?

For this research question, there were fourteen csv files that needed to be cleaned and merged before analyzing - seven about bandwidth and seven about computer science course offerings. To begin, we read in each of the bandwidth datasets, dropped extra formatting in the first few rows, and renamed each of the columns. We then dropped unnecessary columns such as an empty column and the building code column. We checked for and dropped duplicate rows, stripped extra spaces from strings in some columns, deleted rows without school identification codes (dbn). Then, we reset the index. Once this was done for all sets, we merged the datasets to create one data frame organized by 'dbn' with each column holding the bandwidth data for a single school year. The years were labeled by the year in which the school year ended. For example, the school year 2015-2016 became 2016. We then changed the columns for numerical variables to a float data type. Then, we made a scatterplot matrix to see the relationship between different years' bandwidths. We also created histograms by year and box plots to understand the distribution and descriptive statistics of the data.

Next, we took the same cleaning and preprocessing steps with the seven csv files about computer science course offerings. Initially for the merged courses dataset, we dropped all columns except for the number of course offerings for each year. Then, we merged each individual course offering file together to make a merged dataset, organized by dbn, with each column holding data about the computer science course offerings for a single year. We then changed the numeric variables to a float data type. There were some rows from the original file that were “totals” and made it appear as if a school had over 1000 computer science classes, so we dropped that row. We then generated descriptive statistics for each year. We used a scatterplot matrix to see the relationships between computer science class offerings by year. We also created histograms by year and a box plot to understand how the distribution of course offerings changed over time.

Next, we concatenated the bandwidth dataset with the course offerings data set on dbn and year so that each row represented one school in a single year. The rows were arranged by ‘dbn’, while the rows housed the variables regarding computer science classes and computer science course offerings. We then changed the column data types to floats for the numerical variables. Then, we gathered descriptive statistics on all of the variables by year. We created a scatterplot to illustrate the relationship between bandwidth and the number of computer science classes, color coded by year. Next, we created scatterplots of the same data, filtered by year, with fitted regression lines for each year. Finally, we found the correlation between bandwidth and computer science course offerings by year.

Afterwards, we noticed that the values of the correlations were low and negative, which we found strange. The number of schools that had course offering data was low across all years, but especially in the years right after the CS4All law was enacted. This could mean that these schools did not have any CS classes, or it could mean they did not turn in their report. Or it could mean both. We thought it was more likely that they didn’t have any computer science classes because the minimum number of computer science classes in the provided data was 1. It seemed unlikely that every school in New York City has had a computer science course available since 2015. We wanted to see how affected the trends in the data if we assumed that those schools had no computer science classes. Next, we changed all NaNs to zero in the course offering variables. We then ran descriptive statistics, created a boxplot, scatterplot, histograms with regression lines, and correlation matrix to compare the dataset with NaN’s to the dataset with zeros.

RQ4 - Does the number of stem teachers in a school in a given year correlate to the number of computer science classes offered?

For this question, we used steps similar to the ones above when reading in and cleaning the seven csv files about STEM teacher numbers by school. We dropped extra formatting in the first few rows,renamed each of the columns. We then dropped unnecessary columns such as the ‘school name’ and ‘ratio_ftteachertostud’ columns. We checked for and dropped duplicate rows, stripped extra spaces from strings in some columns, deleted rows without school identification codes (dbn). Then, we reset the index. Once this was done for all sets, we concatenated all seven years’ datasets. We then organized and renamed the schools’ types as there were inconsistencies in the existing labels. We then changed the column data types to floats for the numerical variables. We then used boxplots, descriptive statistics, and scatterplots to understand the distribution of STEM teachers across years and school types. We also added three columns, one to show each of the following calculations: the number of fulltime STEM teachers per student, the number of part-time STEM teachers per student, and the total number of STEM teachers per student.

Next, in order to answer the research question, we merged together the ‘bandwidth and course offerings' data set together with STEM teacher datasets, resulting in a large dataset of rows organized by dbn and year, with the following variables: bandwidth, computer science classes, computer science AP classes, computer science full classes, computer science partial classes, number of STEM teachers, number of full time STEM teachers, number of part time

STEM teachers, the number of fulltime STEM teachers per student, the number of part-time STEM teachers per student, the total number of STEM teachers per student, and the school type. The school type column needed to be reformatted, so we categorized the different school types into 5 categories: K-5, K-12, 6-8, 6-12, and 9-12. Then, we used the dbn to add a column to tell the borough of each school.

Finally, we used the dataset to determine the correlations present in the data between STEM teachers and computer science courses offered. Then, we decided to try finding correlations in the dataset after changing any NaN's in the computer science classes variable to zero, as in the previous research question, to see how it would affect the relationships between the variables. We then used descriptive statistics, box plots, and scatterplots to understand the relationships between the variables.

3. RESULTS OR FINDINGS

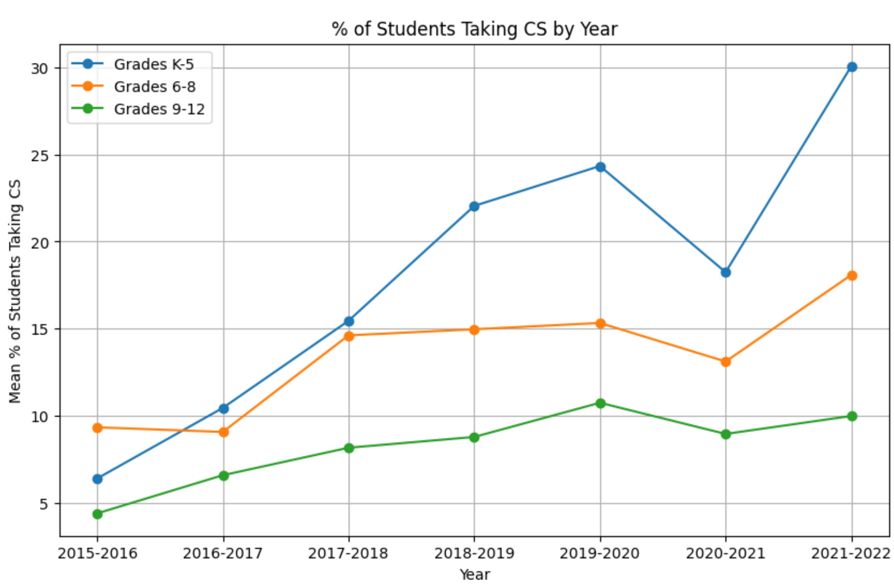
3.1 Results of Research Question #1: How have the demographics of students taking computer science changed over time?

<u>Demographic: Grade Groups</u>

In the analysis of grade groups K-5, 6-8 and 9-12 we had 231 values of data from 2015-2022. The average number of students across the seven year period 29,068.06 with an average of 4021.05 taking a CS class. 2497.67 students were in Grades K-5, 953.41 students were in grades 6-8 and 777.61 were Grades 9-12. Over time the students who took computer science the most were Grades K - 5. We are not sure if these classes are complete courses, AP courses or what the criteria for a school to say a student is taking a computer science course is, but K-5 was and is the largest group learning computer science. On average over the seven year period 13.09% of students took a CS class across all grades.

If you break down the statistics by year you can get a clearer picture of the trend. Overall the average number of students taking computer science increases steadily by year. The average % of students taking CS for all grades in NYC was 4.64 % in 15-16 and doubled in 16-17 to 8.76%, afterwards it had a steady increase each year 12%, 16% and 18% respectively from 17-20. However in 20-21 the year the covid-19 pandemic started the % of students taking CS decreased to 12.6% less than what it was in 17-18. However, this percentage rose again past pre-pandemic percentages in 21-22 to a record high of 18.42% across all grades. The data follows the same trend when you break it down for grade groups.

Figure 1



With respect to grade, the grade taking the most CS classes was K-5 for all years except 15-16 where we had more 6-8th grade students taking CS. In 17-18 K-5 and 6-8 were almost matched in % of students taking CS. From 15-16 to 21-22 the average number of K-5 students taking CS has increased by a factor of 5x from 6.4% to 30%. The average number of 6-8 grade students has almost doubled (1.9x) from 9.35% to 18.09%. For 9-12 grades the number, although being the smallest of the three, more than doubled since 15-16 from 4.4% to 10%.

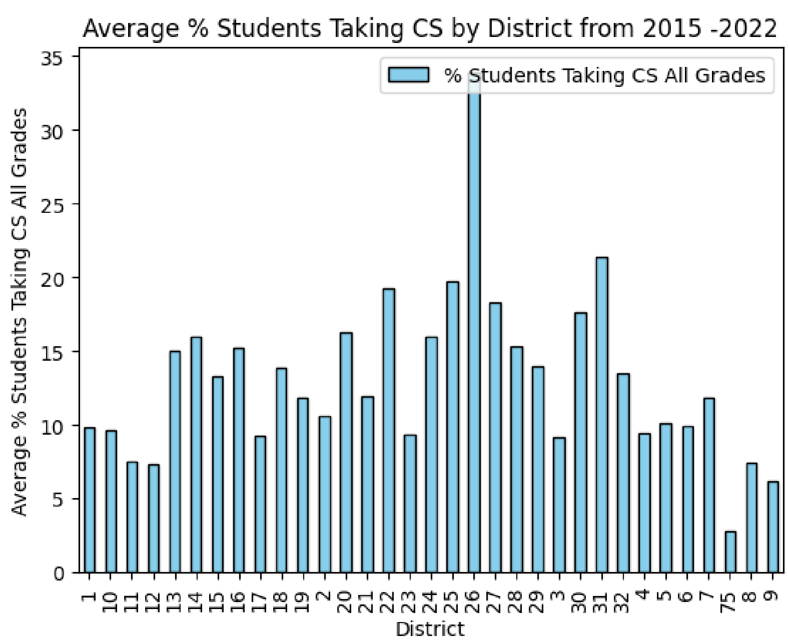
Figure 1 is a Grouped Line Plot highlighting these trends.

<u>Demographic: District</u>

The district with the students taking the most CS classes is District 26 with an average of 33% of students taking CS. The next biggest district is 31 then 25. Note 26 is in Queens, 31 is Staten Island and then 25 is back in Queens. The districts with the lowest average % of students taking CS over the seven year period is District 75 then 12 and 8. District 75 comprises schools with highly specialized instructional support for students with significant challenges. Districts 12 and 8 are in the Bronx. Figure 2 is a bar plot highlighting the average % of Students taking CS by district from 2015 - 2022.

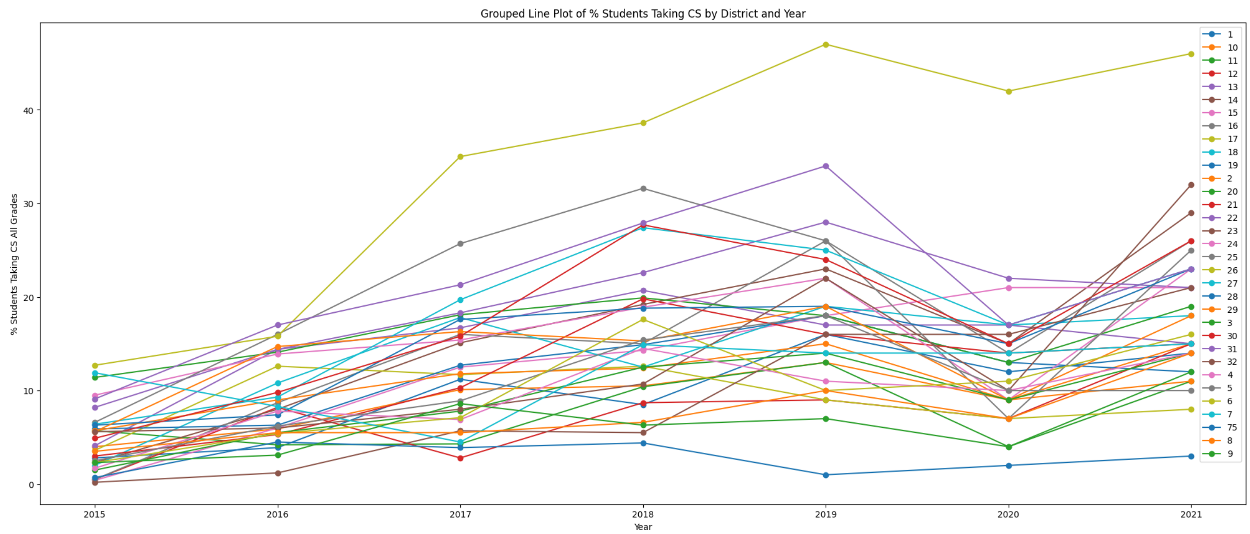
To get a better image of the trends we conducted a regression analysis. When comparing all districts, you can notice that the coefficients vary significantly. Some key points from these results are: Districts with positive coefficients indicate an increase in % Students Taking CS All Grades over the years, while a negative coefficient (such as for Districts 17 and 75) would suggest a decrease. Districts with statistically significant p-values (less than 0.05) have a more reliable relationship between the year and % Students Taking CS All Grades. Districts 1, 3, 4, 8, 10, 12, 14, 15, 18, 21, 22, 23, 28, and 32 all demonstrate statistically significant positive changes, although their rates of change (as indicated by the coefficients) are lower than for District 26. Districts with a large positive coefficient and a low p-value, like Districts 14, 15, 23, and 26, are showing a significant positive trend in student participation in CS classes over the years. Conversely, District 17, even with a negative coefficient, has a very high p-value (0.97), suggesting that the observed negative trend is not statistically significant and might as well be due to random chance rather than a real decline.

Figure 2



For District 26 the coefficient is 5.867 this means that on average in this district, there is an increase of about 5.87 percentage in % Students Taking CS All Grades with each passing year. The large slope signifies a greater rate of increase over the years. The p-value is 0.005. Since the p-value is less than 0.05, it is statistically significant. This indicates that there is strong evidence of a relationship between the years and % Students Taking CS All Grades in District 26. Figure 3 is a Grouped Line Plot of % Students Taking CS by District and Year.

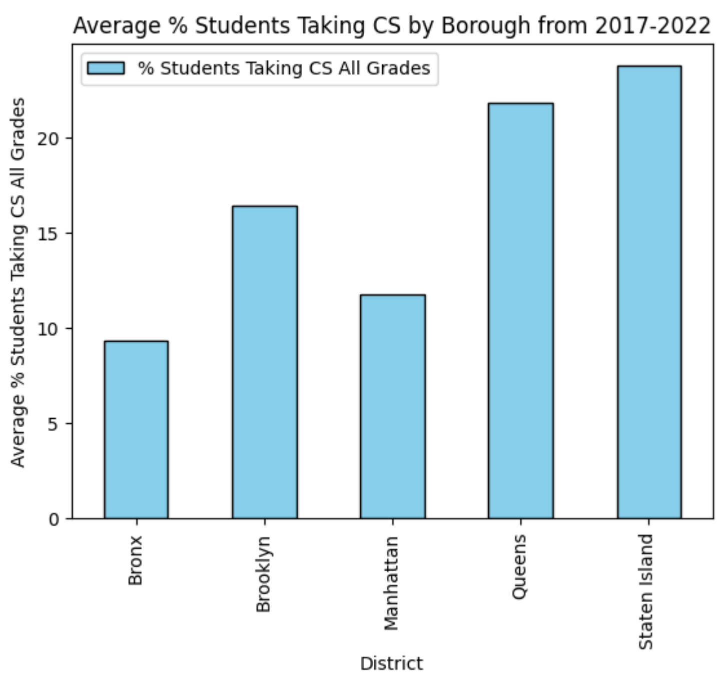
Figure 3



<u>Demographic: Borough</u>

For this Demographic there is no data for 2015 - 2017. As shown in the previous demographic the districts with the most students taking computer science were in Staten Island and Queens. In this borough breakdown we can see similar trends and groupings to the district data.

Figure 4



This could be attributed to the size of the borough, which leads to more schools and students, however the percentages are based off of the entire population of students in that specific borough. We performed a linear regression to see if this immediate deduction still holds.

Figure 5

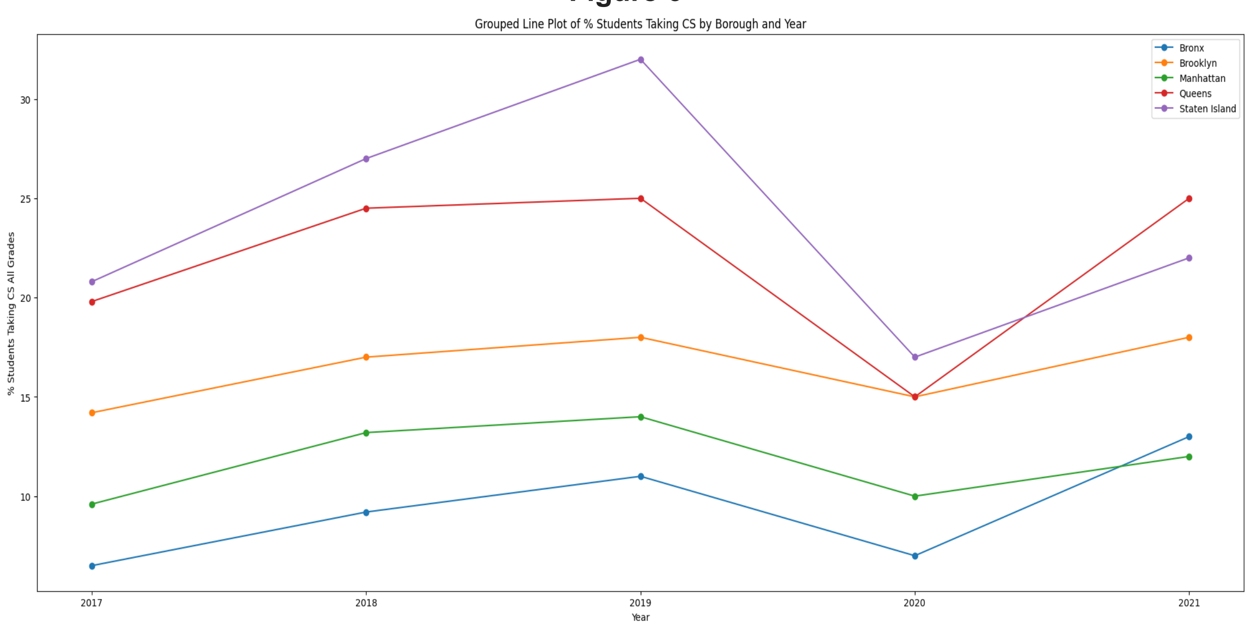
```
Borough with the most significant change over years: Bronx
Coefficient: 1.079999999999788
P-value: 0.25805486987797305

All Borough:
Bronx: Coefficient: 1.079999999999788, p-value: 0.25805486987797305
Brooklyn: Coefficient: 0.5600000000000022, p-value: 0.3849370178658033
Manhattan: Coefficient: 0.1600000000022063, p-value: 0.8336531707234096
Queens: Coefficient: 0.0900000000002476, p-value: 0.958974639395971
Staten Island: Coefficient: -0.7599999999991324, p-value: 0.7394298168627564
```

The coefficient represents the average change in the percentage of students taking CS courses for each borough over the 5-year period. According to the coefficient the borough that has had the most change in % of students taking computer science classes is the Bronx. Each year the percentage of students taking CS courses in the Bronx increases by an average of 1.08% each year. Which is not enough and would suggest more funding should go towards computer science education in the Bronx. Although it's the borough with the smallest average there have been some changes. Based on the p-value results, none of the boroughs show a statistically significant relationship between the change in the percentage of students taking CS courses over the 5-year period and the borough at the 0.05 significance level.

Figure 6 is a grouped line plot showing the relationship between the % of students taking CS by borough across 5 years.

Figure 6



<u>Demographic: English Language Learners (ELL)</u>

Over the 7 year period from 2015 - 2022 the divide between ELL and Non-ELL students taking computer science is almost the same with an average difference of 0.4. We needed to pay close attention to the trend and any changes from year to year. Upon initial observations of the descriptive statistics the mean non - ELL students have a higher % of students taking CS from 2015 - 2020. However, it seems in 2020-2021 the average number was 50/50 and then the trend flipped. In 2021 - 2022 now we have more ELL students taking computer science (average 22%) vs Non-ELL (average 18%). We would have to do more research to see why there was a change. We ran a regression to see if it is statistically significant.

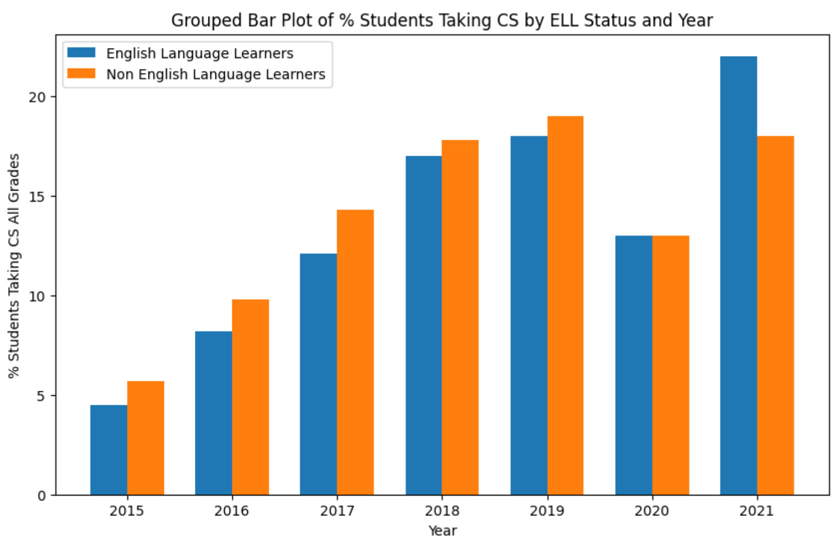
Figure 7

```
ELL Status with the most significant change over years: English Language Learners
Coefficient: 2.4285714285714564
P-value: 0.010041858952375526

All Districts:
English Language Learners: Coefficient: 2.4285714285714564, p-value: 0.010041858952375526
Non English Language Learners: Coefficient: 1.714285714285725, p-value: 0.04811140067001811
```

ELL (English Language Learners) status shows a statistically significant change over the years. both p -values are less than 0.05 indicating the effect of ELL status on % of students taking CS is statistically significant. The coefficient of 2.428 indicates there has been a large increase in the average percentage of ELL students taking computer science. The grouped bar plot below visually shows this trend.

Figure 8



<u>Demographic: Economic Status</u>

Similar to ELL the mean % of students who are economically disadvantaged was lower than students who are not economically disadvantaged. Both percentages dropped during 20-21 and then increased but flipped, meaning in 21-22 we have more economically disadvantaged students taking CS by an average of 1 %.

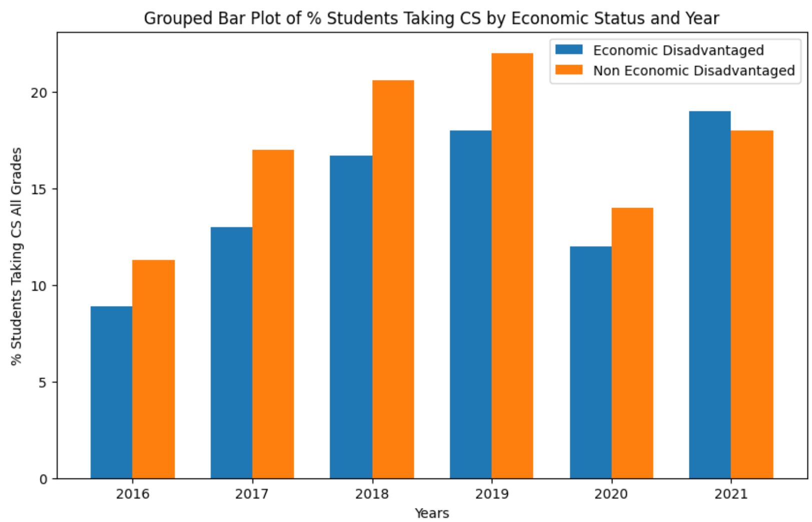
Figure 9

```
Economic Status with the most significant change over years: Economic Disadvantaged
Coefficient: 1.3942857142857257
P-value: 0.15029681591555516

All Districts:
Economic Disadvantaged: Coefficient: 1.3942857142857257, p-value: 0.15029681591555516
Non Economic Disadvantaged: Coefficient: 0.7399999999999798, p-value: 0.502471707136964
```

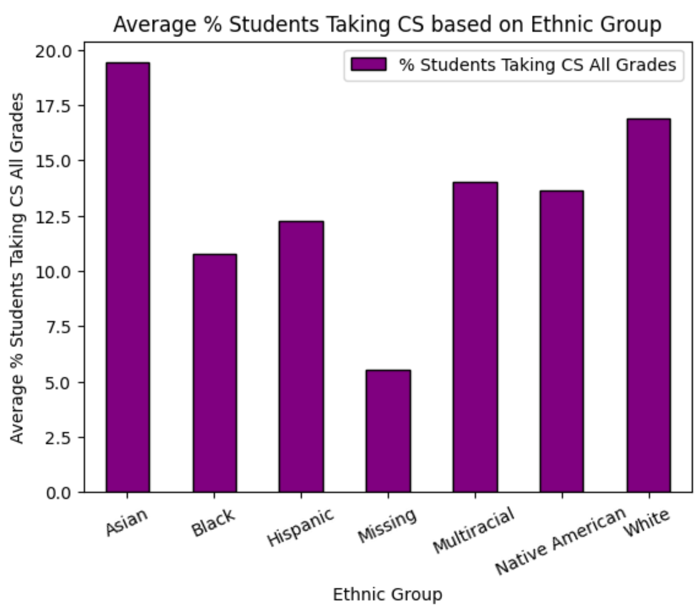
The economically disadvantaged group (Coeff: 1.39) has shown a more significant increase in the percentage of students taking CS over the years compared to the Non-Economic Disadvantaged group (Coeff: 0.7399). This suggests that economic disadvantage has a larger effect on the increase in CS participation over time. The P-Value of 0.150 means there is a moderately significant relationship between economic disadvantage and the change in CS participation over the year. Important to note this study is missing information from 2015-2016.

Figure 10



Demographic: Ethnicity

Figure 11



The ethnicity with the highest average % of students taking CS are Asians, 2nd highest white, multiracial and then native american. We conducted a regression to gauge statistical significance.

Figure 12

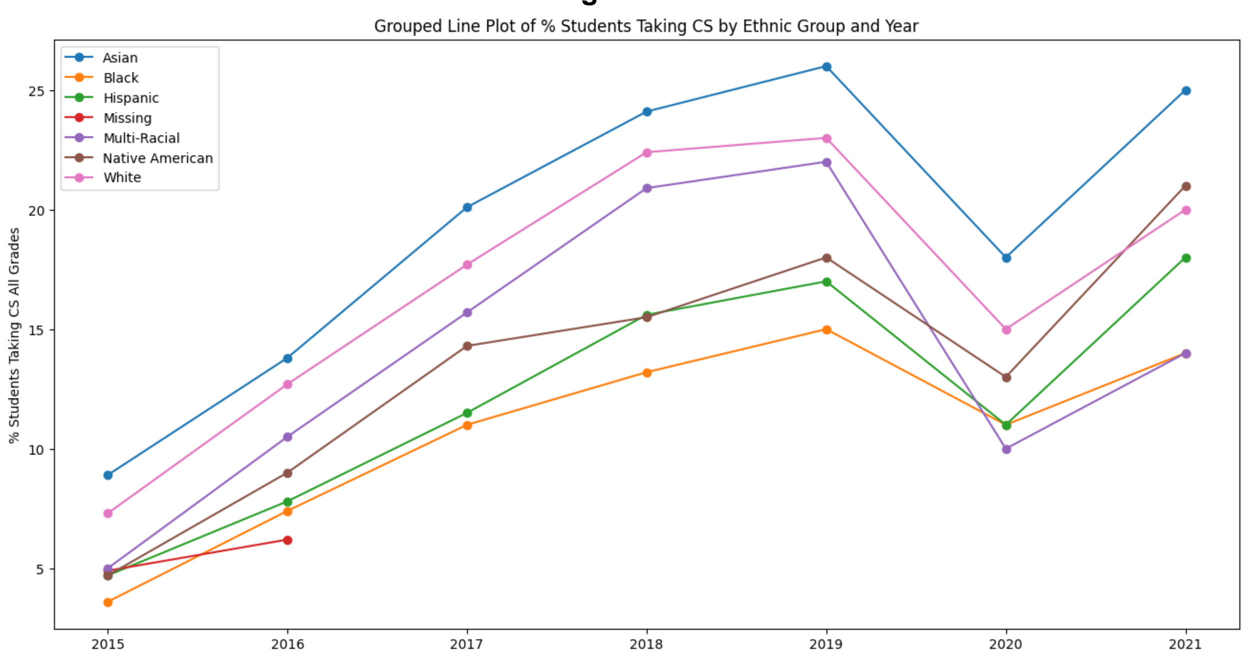
```
Ethnicity Status with the most significant change over years: Asian
Coefficient: 2.2357142857143004
P-value: 0.04649205838118352

All Districts:
Asian: Coefficient: 2.2357142857143004, p-value: 0.04649205838118352
Black: Coefficient: 1.5142857142857242, p-value: 0.026129735223168776
Hispanic: Coefficient: 1.8500000000000183, p-value: 0.026933406416781408
Missing: Coefficient: 1.29999999999978, p-value: nan
Multi-Racial: Coefficient: 1.1535714285714191, p-value: 0.3634371870158671
Native American: Coefficient: 2.164285714285735, p-value: 0.013931771131604058
White: Coefficient: 1.7142857142857184, p-value: 0.10945843741333304
```

The analysis shows that the Asian ethnic group has the most significant change in the average percentage of students taking CS over the years, with a coefficient value of 2.235. This shows a big increase in CS participation within the Asian community. The p-value of 0.0464 for the Asian ethnic group suggests that there is a statistically significant relationship between Asian ethnicity and the significant change in CS participation. The low p-value means a high level of confidence in the relationship observed.

When comparing the change in CS participation across all ethnic groups, the coefficients and p-values for the Black, Hispanic, Native American, and White ethnic groups also indicate significant changes in CS participation over the years. Note that for the "Missing" ethnic category, the p-value is labeled as "not a number." This indicates missing or invalid data which we narrowed down to be from 2015-2016. These results provide valuable insights into the changes in CS participation among various ethnic groups.

Figure 13



This graph provides a bit more detail regarding the change over time. All ethnic groups had a steady increase. All groups % of participation in CS class dropped in 20-21. It is important to note that the multiracial group who was ranked 3rd dropped to the least % in 20-21. Black and multiracial student % of participation increased but were almost the same in 21-22. Native American students shot up to 2nd most in 21-22 from 3rd and 4th in previous years.

Some potential reasons for these fluctuations could be educational policies and programs, shifts in the availability of CS courses, variations in the representation and encouragement of different ethnic groups in STEM fields. Population shifts could also contribute as the pandemic saw a large fluctuation in population moving in and out of NYC.

<u>Demographic: Gender</u>

The average % of students taking CS based on gender is almost the same but there are a few more male students taking CS. Both coefficients are really close and somewhat significant. Female students have a slightly lower change in CS participation compared to male students.

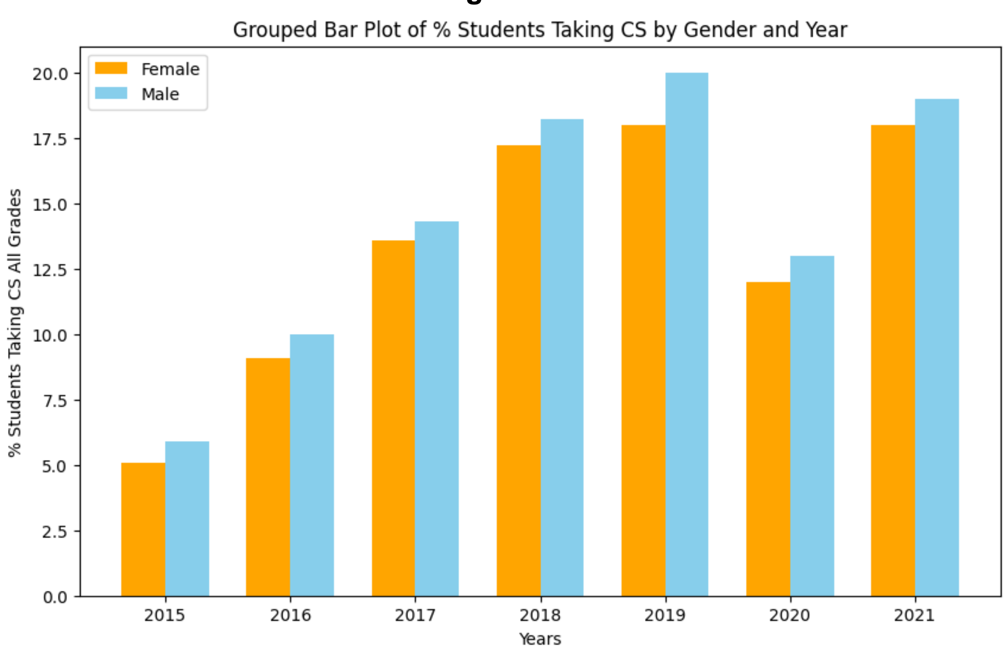
Figure 15, the grouped bar plot gives us a better idea of the trend for each year, the other demographics had a comeback and surpassed pre-pandemic (2020-2021) numbers but gender did not.

Figure 14

```
Gender Status with the most significant change over years: Male
Coefficient: 1.8214285714285843
P-value: 0.04705543150789996

All Districts:
Female: Coefficient: 1.7464285714285834, p-value: 0.04524014223741193
Male: Coefficient: 1.8214285714285843, p-value: 0.04705543150789996
```

Figure 15



Demographic: Disability Status

On average there are a bit more students taking CS without disabilities than with a disability. The bar plot below shows a similar trend to gender which means the relationship between sub demographics (male v female, with or without disability) has not changed much but the overall % of students taking CS has increased.

Figure 16

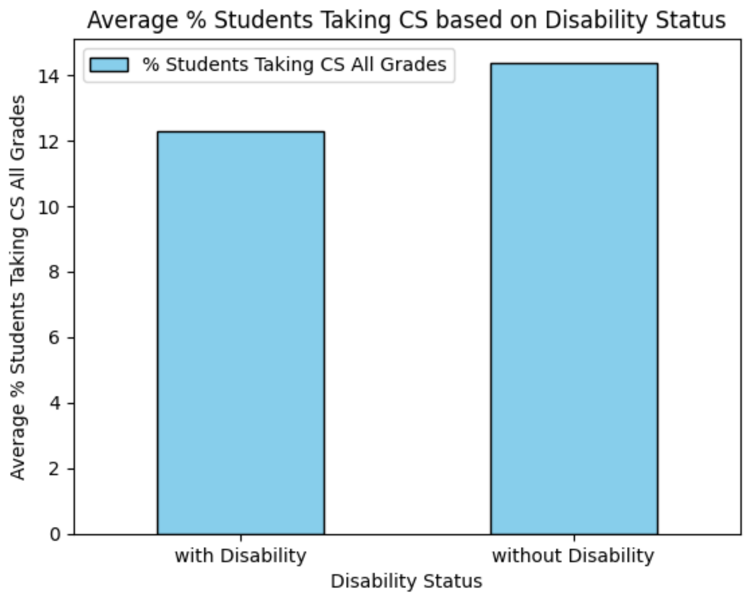
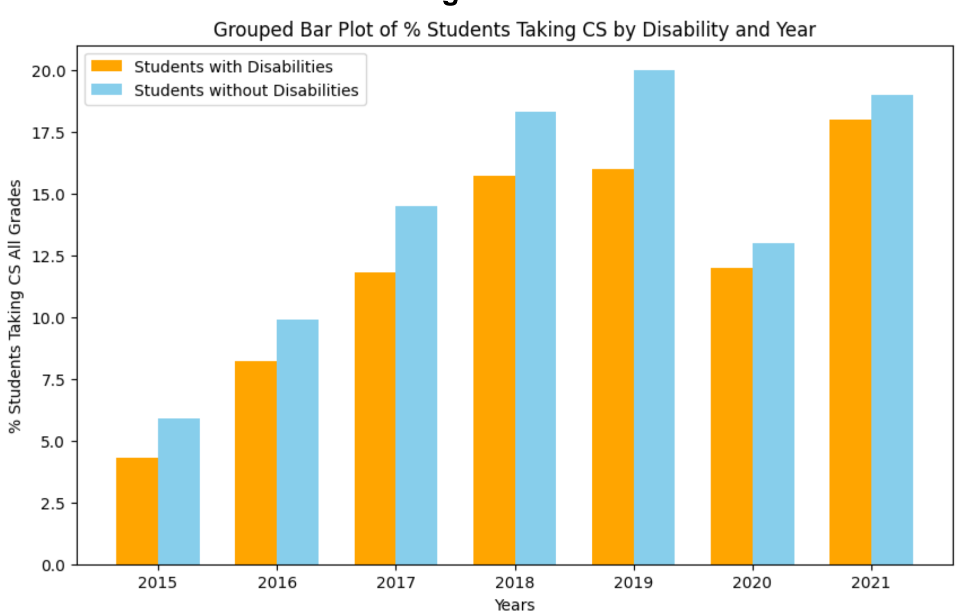


Figure 17



3.2 Results of Research Question #2: Do computer science course offerings differ by borough?

Over the years there have been more and more computer science classes. There are 69 Not A Number Values which might affect the results. Based on descriptive statistics there are more computer science classes than AP courses. At a glance the borough with the most schools is Brooklyn. The borough with the least number of schools is Staten Island (SI) but SI has the most computer science classes. This tells me that the enrollment in these courses is fairly large or a lot of schools did not report their class.

Figure 18

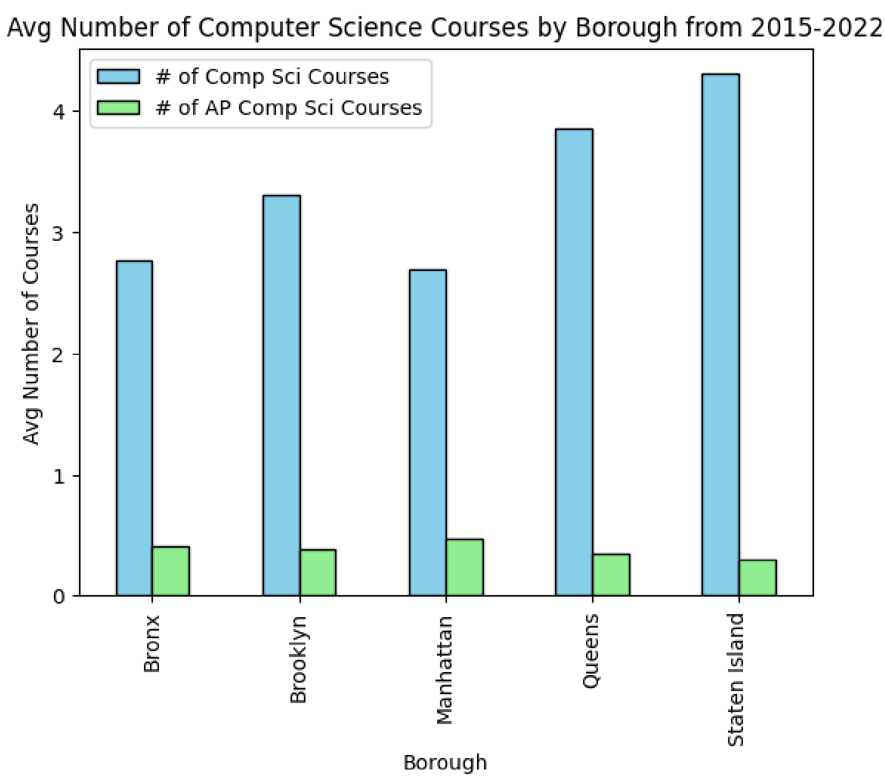
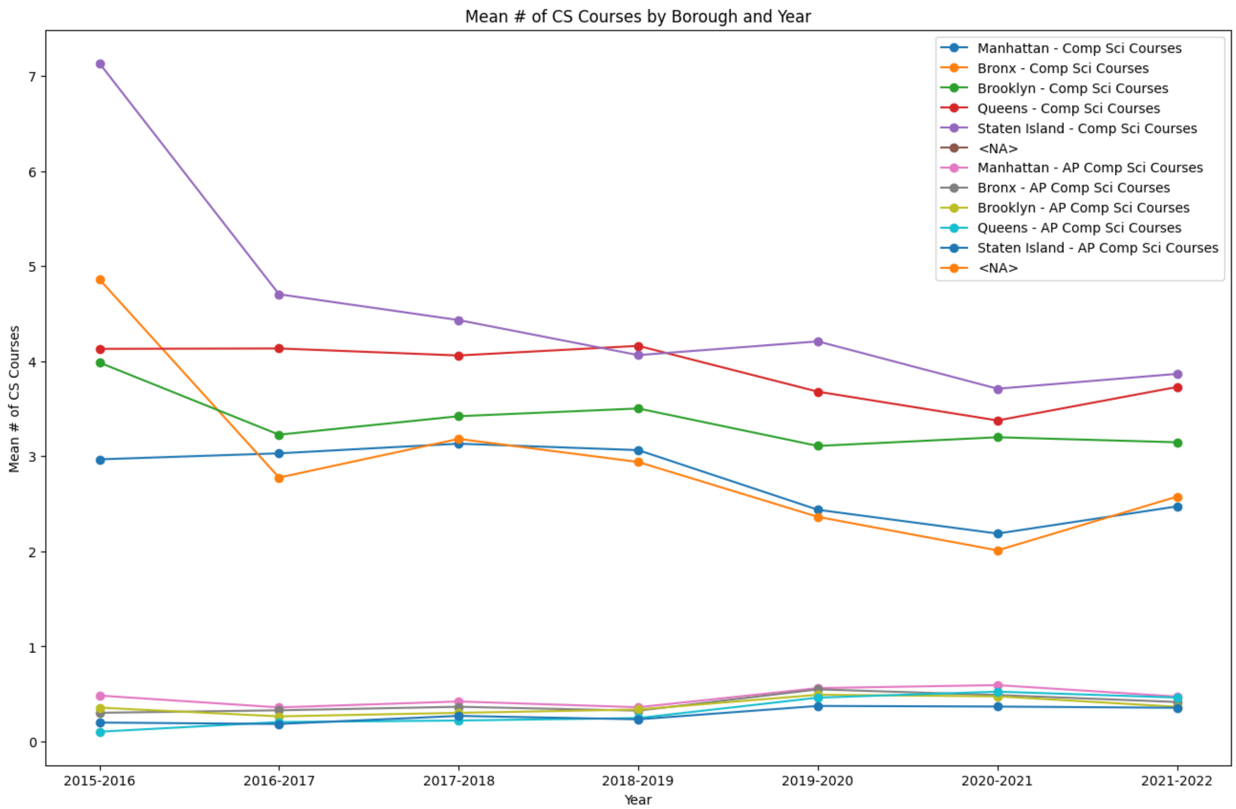


Figure 19



As shown by Figure 19 the average number of computer science classes has gone down over the years. The classes differ by borough but they have for the most part decreased in quantity. We have not reached pre-pandemic numbers (19-20) as of yet in 21-22. There are many possibilities for this. We have seen the trend of more students in computer science classes but it seems the average course amount has stayed the same. Perhaps classes are overfilled or there is some computer science education in schools but not in a formal capacity or these schools are underreporting the number. It could also be that some schools with 0 classes are pulling down the average number as a whole. The average of AP computer science courses has fluctuated but not by much.

3.3 Results of Research Question #3: What is the relationship between school bandwidth and computer science course offerings?

Figure 20 is a boxplot of the bandwidth per school per year. It shows that the bandwidth per school medians have increased between 2016 and 2022 from 20 to 300. It also shows that the biggest median jumps in bandwidth occurred in the 2017-2018 school year and in the 2021-2022 school year. The range for 2022 is 0, which shows that there are pretty uniform rates of bandwidth across New York City Schools in the 2022-2023 school year. An interesting finding is that from 2016-2019, no schools reported their bandwidth as zero. However, from 2020-2022, a small number of schools reported their bandwidth as zero.

Figure 21 shows a boxplot of the number of computer science classes offered per NYC public school per year. Figure 22 shows the distributions of the same data by year. Figure 23 shows a year compared to year scatterplot matrix comparison of the same data. Because the maximum number of computer science classes offered during the 2015-2016, the schools that had over ten computer science classes the first year were offering under ten by 2022. Figures 21-23 all show that the median number of courses offered among NYC schools has stayed constant at 3 since 2015, which is what the available data indicates. However, the minimum number of courses for all years is 1, which seems unlikely. At this point, we wondered whether the missing computer science courses data from all years, which is close to 1,000 schools in earlier years of data collection, are actually schools that didn't have any computer science classes rather than schools that did not report their computer science classes. This would make a huge difference in some outcomes if true. In later steps, I ran descriptive statistics with all of the NaN's in the computer science courses data set as zeroes in order to see how it affected the relationships between variables.

Figure 20
Boxplot of bandwidth per school by year

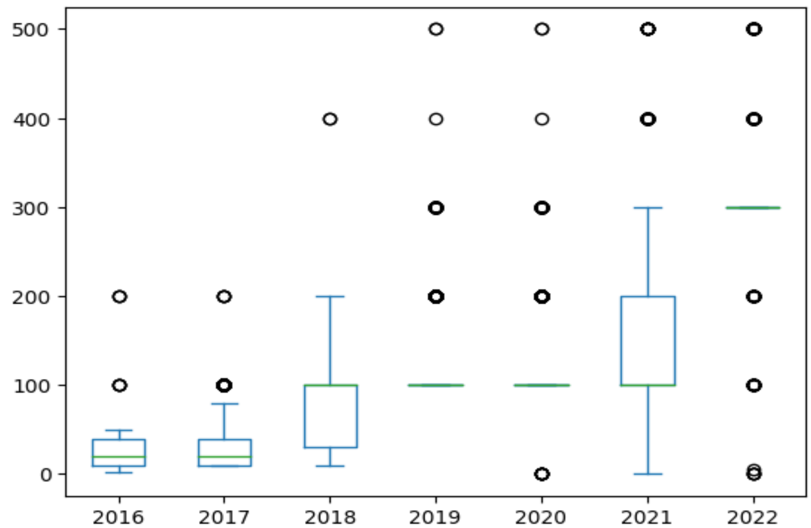


Figure 21
Box Plot of the Number of Computer Science Classes Offered at Each NYC Public School

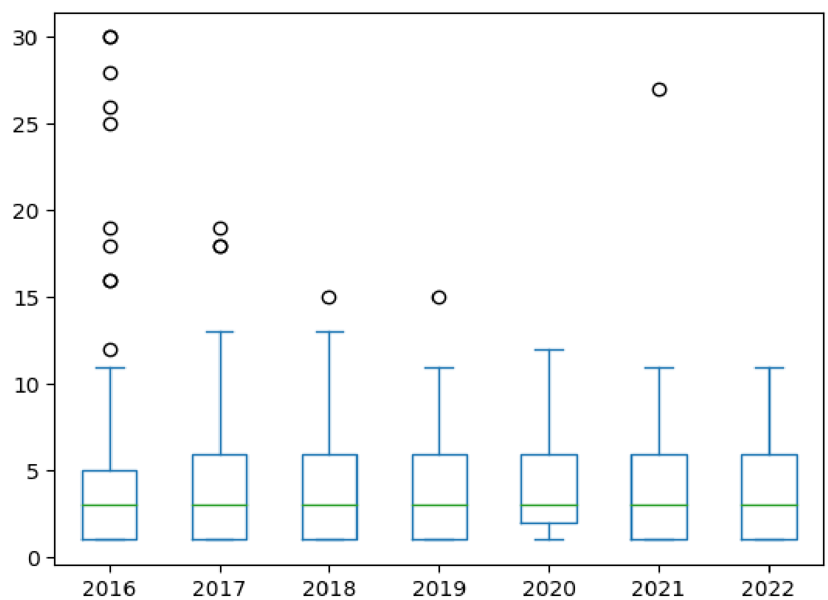


Figure 22

Histograms of Computer Science Courses Offered per NYC Public School by Year

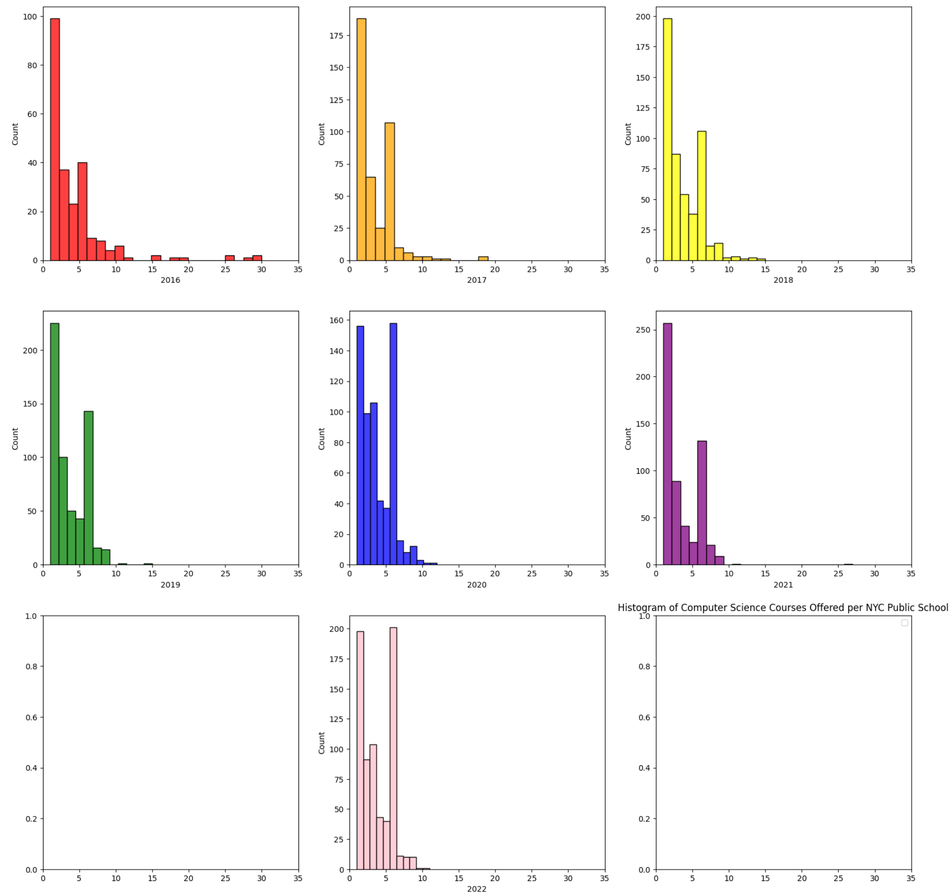


Figure 23

Scatterplot of Computer Science Courses Offered by Year Compared to Year

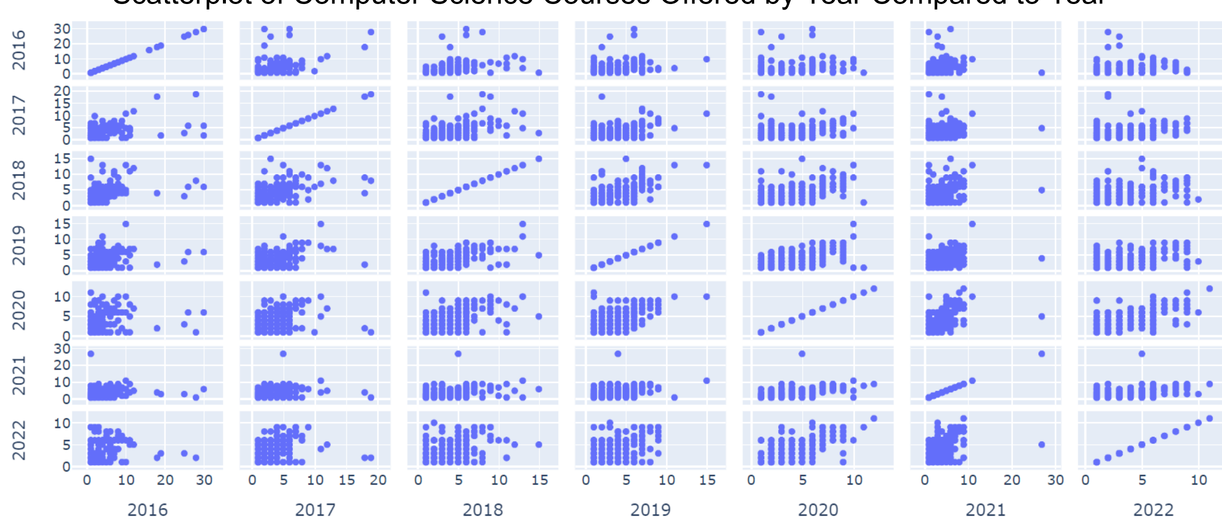


Figure 24

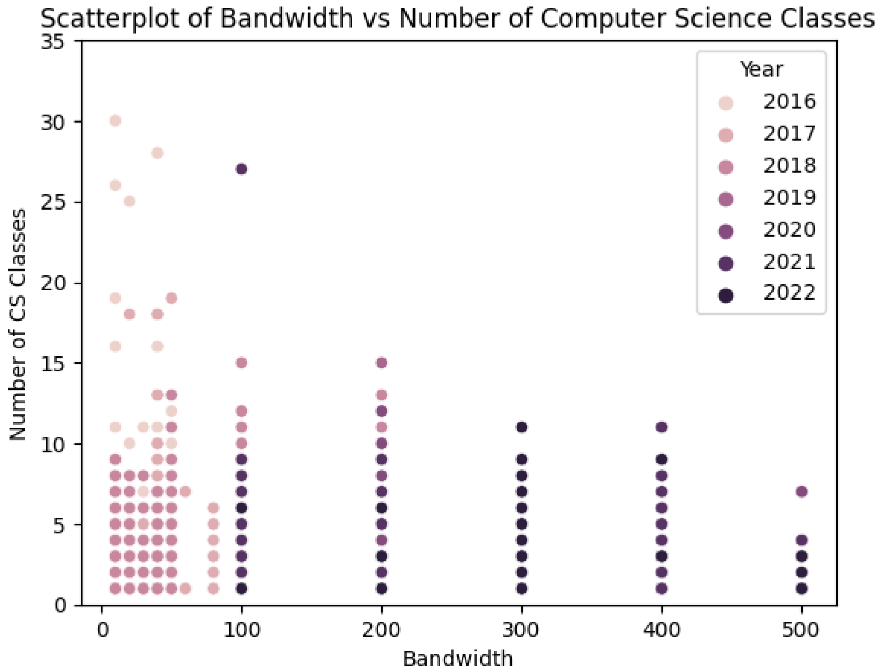


Figure 24 explores the relationship between school bandwidth and the number of computer science courses offered by year. It is interesting that both the schools' bandwidths and the number of computer science classes they offered became more centralized by 2022, but were more spread out in 2016. This leads to the seemingly negative relationships shown in the scatterplots with regression lines in Figure 25. Thus, as a school's bandwidth decreased, the number of computer science courses they offered decreased. The correlation between the variables for each year was: -.08, -.16, -.16, -.17, -.14, -.13, and -.04. We were curious about this relationship since many schools, especially in the years early after the law was enacted, did not have data in the computer science courses dataset. This could mean that they didn't have any CS classes, or it could mean they didn't turn in their report. Or it could mean both.

Figure 25

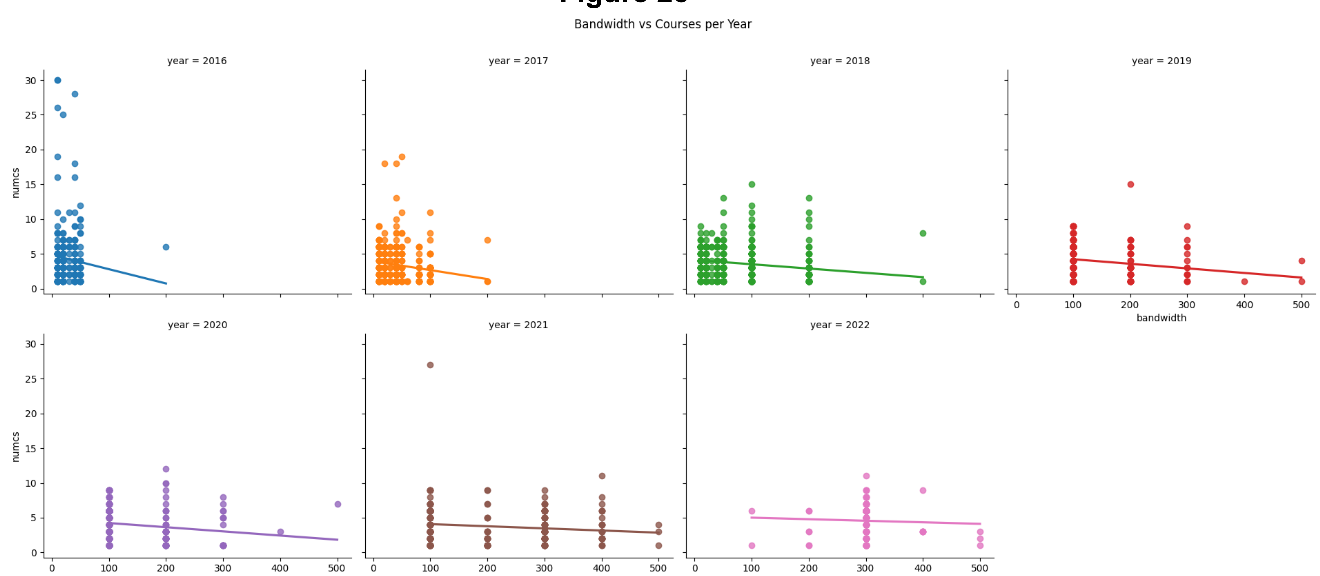


Figure 26

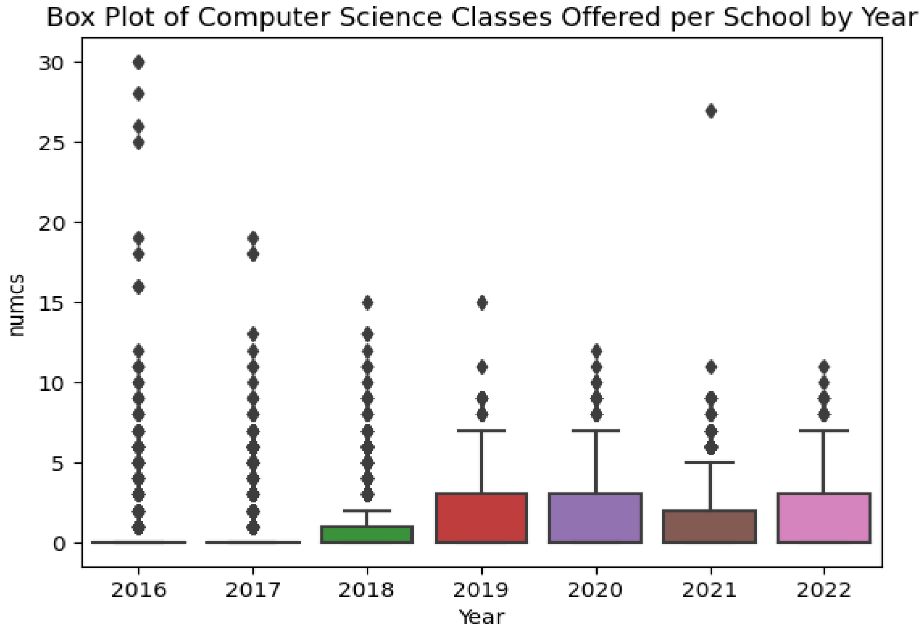
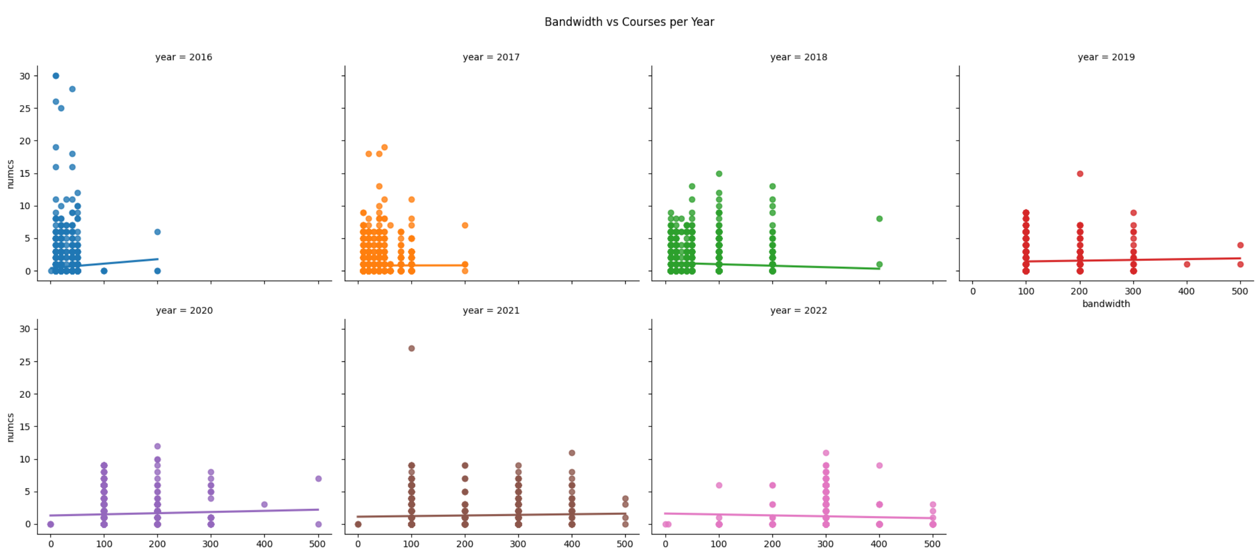


Figure 27



In order to determine how the relationship between the variables would change if the NaN's in the CS course datasets were actually indicative of having no compute science courses, I changed all NaN's in that specific dataset to zeros and got the descriptive statistics, correlation, and scatterplots with regression lines. As Figure 26 indicates, the median number of classes and the entire range then go down to zero in 2016 and 2017, with the schools that have computer science classes all as outliers. The range of classes from 2019 to 2022 is on average between 0 and 6. The regression lines for each year then seem more level, indicating that there is likely no relationship between a school's bandwidth and the number of computer science courses they offer. The correlations between the variables by year changed as well. They are, respectively, as follows: .05, .001, -.06, .02, .03, .04, -.04.

3.4 Results of Research Question #4: Does the number of stem teachers in a school in a given year correlate to the number of computer science classes offered?

Figure 28 shows two boxplots. The boxplot on the left shows the number of full-time STEM teachers per school by year. The boxplot on the right shows the total number of STEM teachers, which is the number of full-time plus part-time STEM teachers, per school by year. The charts are very similar, reflecting the fact that there aren't many part-time STEM teachers in NYC public schools. The median number of total STEM teachers (2-3) and the mean number of STEM teachers (5.14-5.39) stays relatively constant over the years of the study. The ranges stay rather constant, however, the Q3 number drops a little in the 2020-2021 and 2021-2022 school years. This could be a reflection of the nationwide trends of high numbers of teachers leaving the field since the COVID-19 pandemic, along with a nationwide shortage in STEM teachers. It could also be a reflection of the reducing number of students in public schools. The mean number of students per school from 2016-2021 ranged between 589 and 607. However, in the 2021-2022 school year, that number dropped to 536. The median number of students dropped similarly, ranging from 463 to 486 from 2016-2021 and dropping to 429 in 2021-2022.

Figure 28

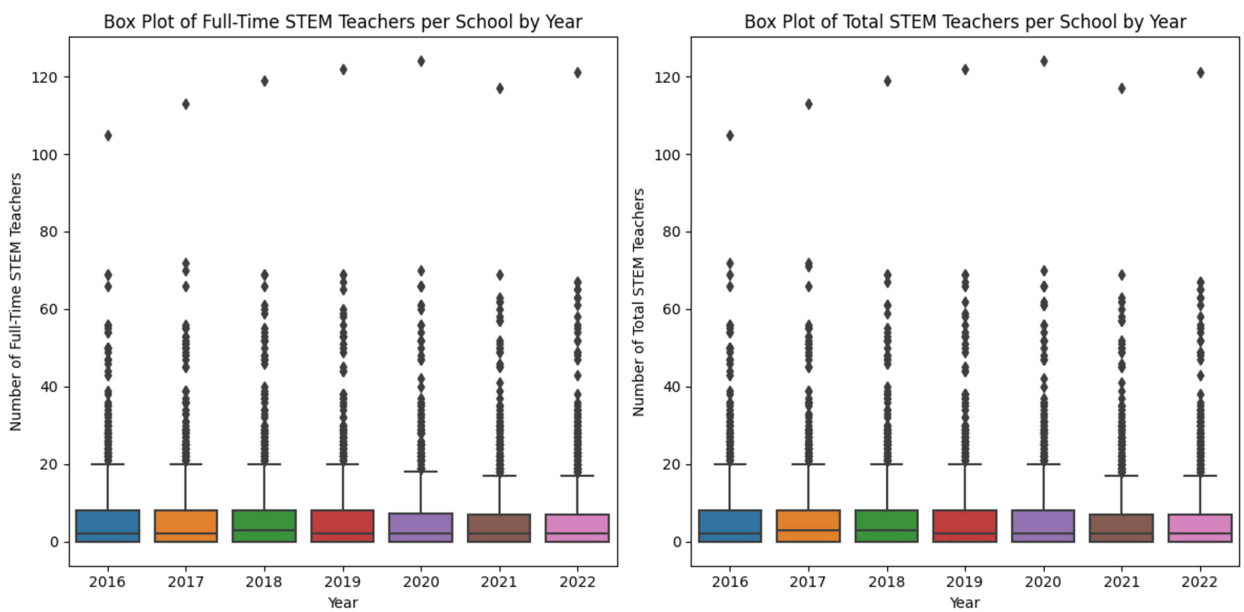


Figure 29 shows a line graph depicting the number of schools in New York City public schools by grade band per year. There are more elementary schools in New York City than other types, followed by high schools, middle schools, and then the variety of schools that crossover grade bands. Figure 30 shows a boxplot of the total number of STEM teachers per school by school type. On average across all years, high schools have the highest number of STEM teachers (11.47), followed by secondary schools serving grades 6-12 (10.83), middle schools serving grades 6-8 (8.96), K-12 all grades schools (3.99), K-8 schools (2.80), K-5 (0.086), and Pre-K (0.00). This is not surprising because older grades teach STEM classes as full stand-alone courses, requiring dedicated teachers with a teaching license in a science or math discipline, whereas those subjects are embedded in the Elementary teacher's license. Therefore, schools serving higher grades have more STEM teachers.

Figure 29

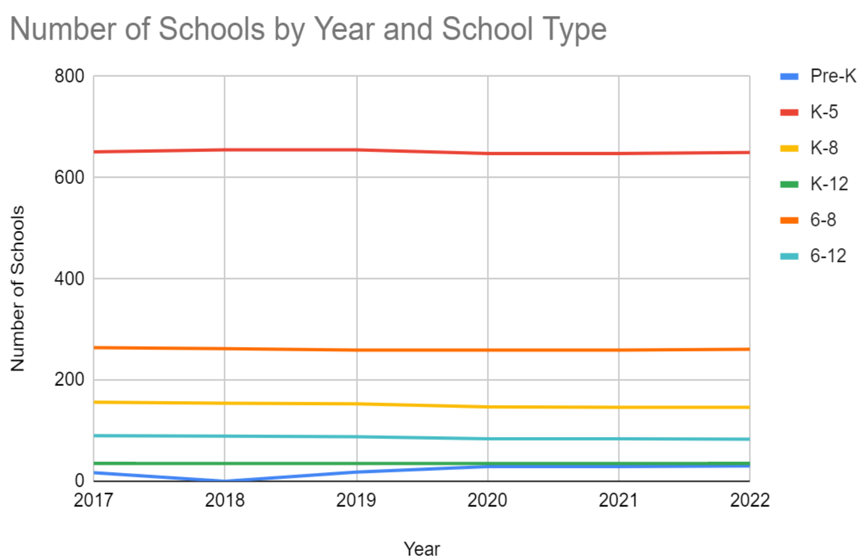
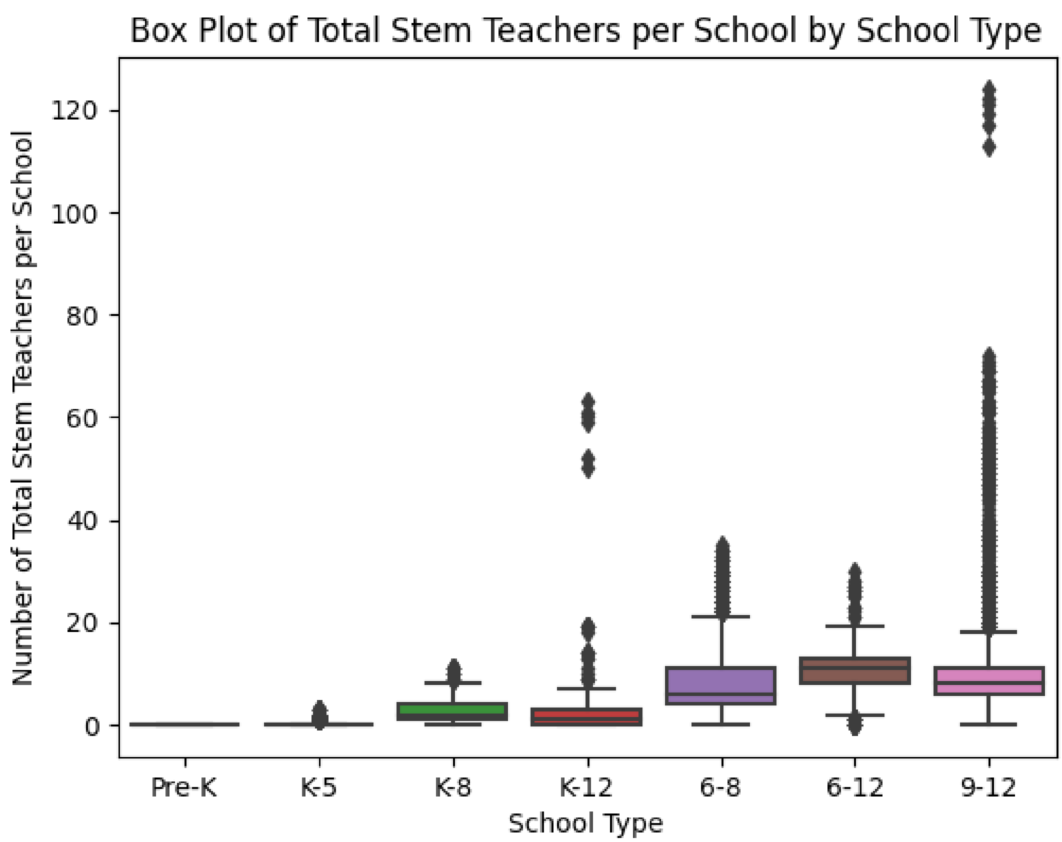


Figure 30



There is a strong correlation (0.715) between the number of students in a school and the number of FT STEM teachers. According to this data, there is no linear relationship (-.09) between the number of computer science courses and the number of fulltime STEM teachers. However, there is a moderate positive relationship between the number of AP computer science courses and the number of fulltime STEM teachers (.511). I assume this is because high schools have both higher numbers of STEM teachers and are the only grade band where students take AP classes. There is a low positive correlation (.31) between the number of fulltime STEM teachers at a school and the number of full computer science courses. The relationship is the inverse (-.29) between the number of fulltime STEM teachers and the number of partial computer science courses. This may be due to the fact that elementary school teachers do not usually have STEM specialist teachers and instead infuse computer science courses in the main content of the general education classroom.

After reviewing the relationships between the variables, we, again, wanted to see how the large number of NaN’s in the computer science courses dataset impacted the relationships between the variables if the missing data points were in fact indicating that those schools had no computer science courses. When the course NaN's were converted to zeros, the relationships in the data changed. There was a low to moderate relationship between the number of students at a school and both the number of computer science classes offered (.21) and the number of AP computer science courses offered (.33). The number of fulltime STEM teachers at a school had a positive moderate relationship with the number of full computer science classes offered at a school (.30) and maintained a moderate, though lower relationship, with the number of AP computer science courses offered (.48).

Figure 31

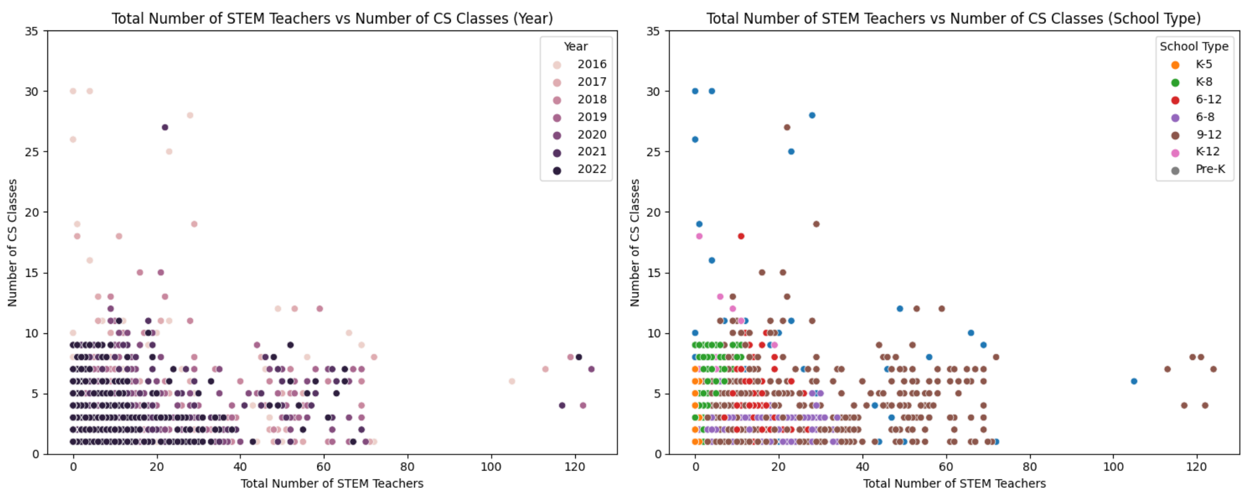


Figure 31 shows two scatterplots of the total number of STEM teachers at a school vs. the number of computer science classes offered using the original dataset with NaN’s. The graph on the left shows it color coded by year, whereas the one on the right is color coded by school type. We can see from the graph on the left that as the number of courses per school became more centralized over time, the number of STEM teachers has stayed rather constant. From the graph on the right, we can see that there are definitely school type clusters in the data. The colors most frequent from left to right also reflect the age order of the schools, with high school teachers tending to have the highest number of STEM teachers.

Figure 32

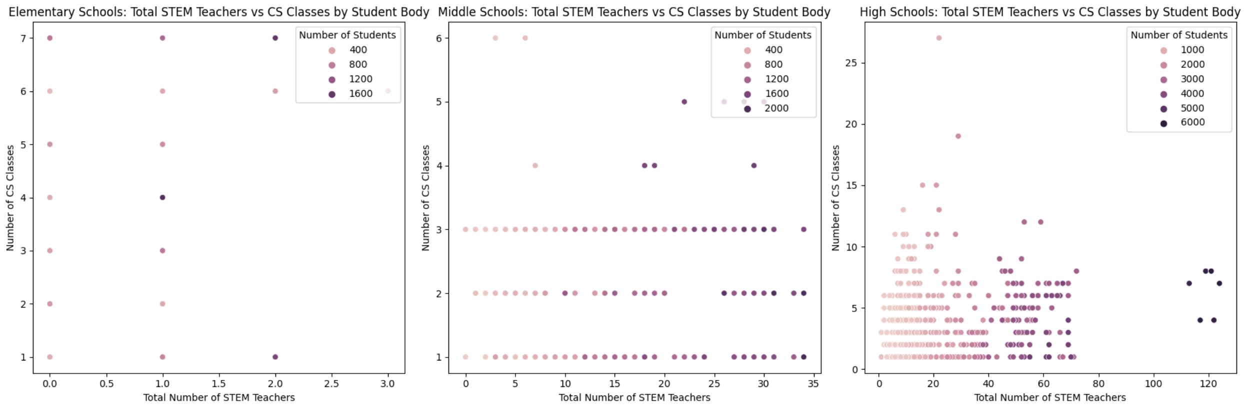


Figure 32 shows the total number of STEM teachers compared to the number of computer classes for elementary schools, middle schools, and high schools. For the purposes of this graph, elementary schools only included grades K-5, middle schools only included grades 6-8, and high schools only included grades 9-12. The elementary school graph has limited data due to the number of missing data points in the computer science courses dataset. Additionally, for reasons discussed before, most elementary schools do not have STEM teachers. However, the middle school plot shows that the number of STEM teachers grows as student numbers increase, though the number of computer science courses, on average, stays the same. The high school plot shows that larger student bodies have more STEM teachers, but not necessarily more CS classes, which is surprising. This leads us to wonder whether the existing computer science classes in middle and high school are at full capacity or not. We also wonder what requirements the computer science classes fulfill, or do not fulfill, and how they are being recommended to students.

Figure 33

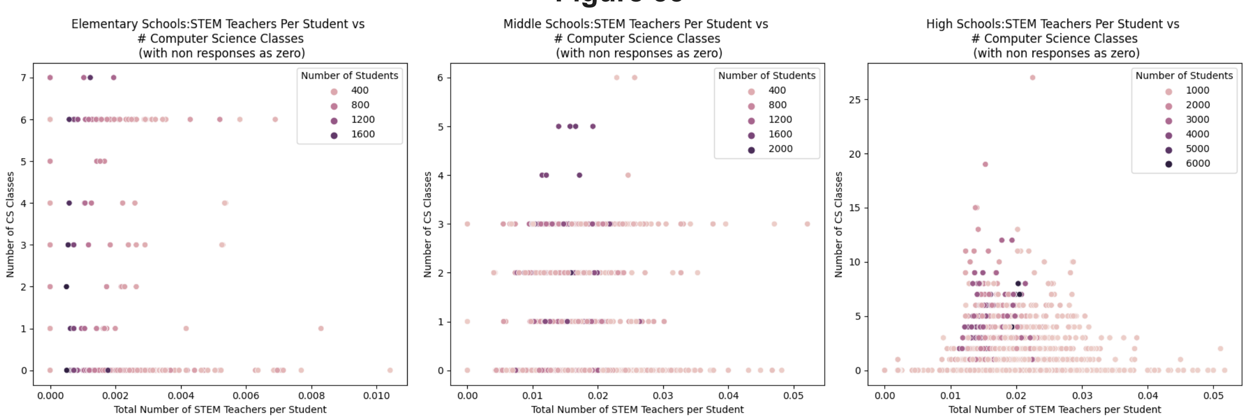


Figure 33 shows the relationship between the total number of STEM teachers per student at a school and the number of computer science classes offered at the school, using the dataset where the NaN's are changed to zeroes. The graphs, from left to right, show elementary, middle, and high school data. The high school graph shows that schools with larger student bodies may have the same number of computer science classes as schools with smaller student bodies, resulting in less total stem teachers per student. We can also see that the data for STEM teachers per student at the high school level is relatively normally distributed. Schools with no computer science classes at the middle and high school level tend to have smaller student populations and have a large range (0-0.05) of total number of STEM teachers per student.

Figure 34

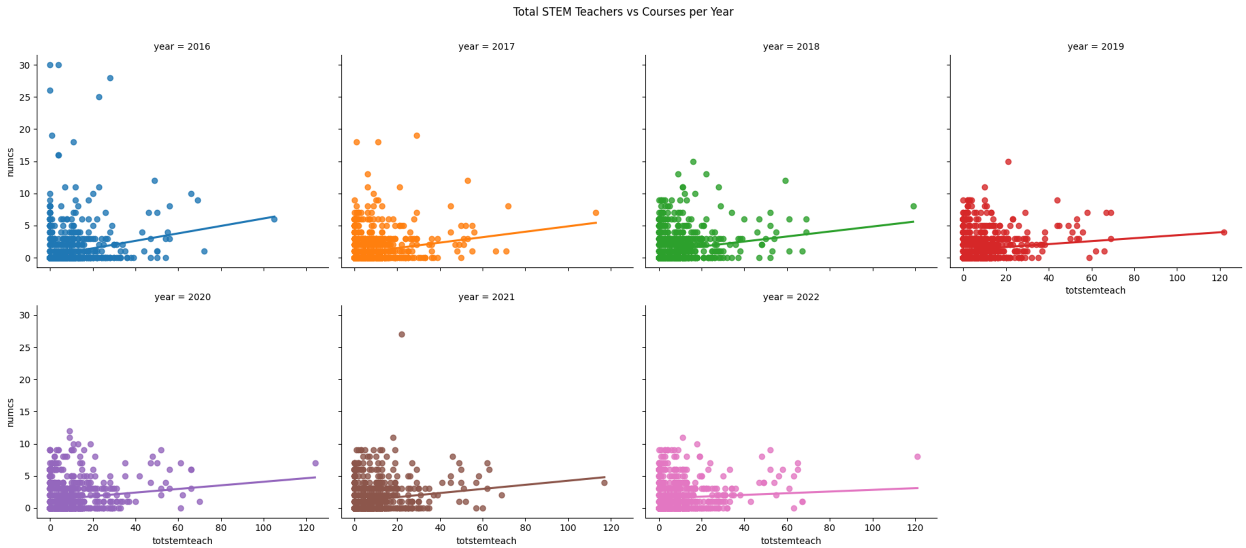
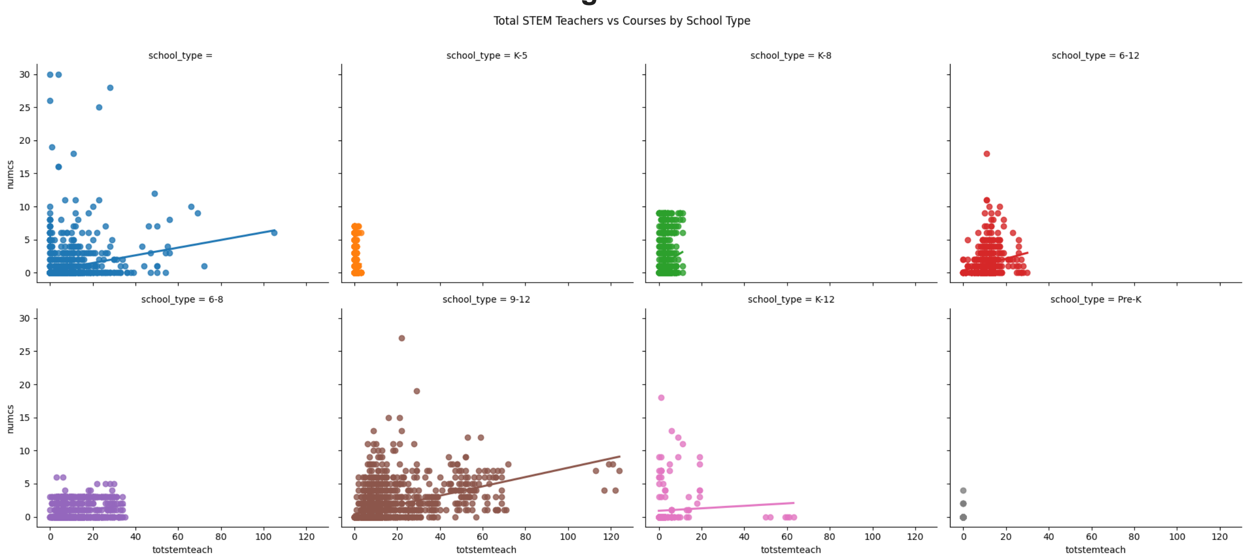


Figure 35



For the scatterplots with linear regression lines in Figures 34 and 35, we used the dataset with NaN's replaced by zeroes. Looking at the scatter plot in Figure 34, which is filtered by year, you can see year after year a positive relationship between the total number of stem teachers and the number of computer science courses. However, none of the lines seem to fit extremely well. The correlations between the variables by year are, respectively: .21, .18, .16, .09, .11, .13, and .05. The scatterplots in Figure 35 show the same data filtered by school type. The relationship at the high school level is the easiest to see. It seems that in high schools, the number of STEM teachers does have a positive linear relationship with the number of computer science classes offered. The elementary and pre-k graphs are not very useful, because a vast majority of schools in those grade levels do not have any computer science classes. The graph in blue is the school type data from 2016, which did not have school types defined.

4. Discussions and Conclusions

Our study shed light on several key trends and dynamics within computer science (CS) education in New York City over the past seven years. Our primary research questions explored demographic changes in computer science enrollment, disparities in course offerings by borough, the effect of school bandwidth on CS course offerings, and the influence STEM teacher numbers have on the availability of a computer science class.

Main Findings:

Demographic Changes: There has been a consistent rise in computer science enrollment across all student demographics since 2020-2021, with a noticeable uptick among K-5 students, English Language Learners, and economically disadvantaged students. The pandemic contributed to a decrease in computer science classes in 2020-2021 across all demographics, but those numbers have since increased past pre-pandemic numbers for most groups. These changes indicate a more inclusive reach of CS education among traditionally underrepresented demographics.

Course Offerings by Borough: Interestingly, despite the evident surge in student enrollment in CS courses, there was a noticeable decrease in the number of available CS courses over the past seven years. The classes stabilized so far with an average offering of 3-4 classes by borough. This discrepancy highlights a potential gap between the supply and demand of CS education within the educational institutions in the study.

Bandwidth and Course Offerings: There is no direct correlation between the bandwidth capacity of a school and the number of computer science courses offered, suggesting that other elements are at play in determining course availability.

STEM Teacher Influence: For high schools, there is a positive correlation between the number of STEM teachers and the availability of computer science courses, indicating that having more STEM educators might positively affect course offerings.

Future Research Directions

The study identified crucial areas for further investigation to enrich the understanding of computer science education. The gaps in data due to insufficient reporting by schools present an imperative need for more comprehensive research. A more longitudinal study will be needed to see the effect of the current initiatives on computer science education. Our analysis covered seven years, but we recommend a reassessment at either 10 or 15 years. It will be prudent to identify how current advancements in technology (i.e., AI, EdTech, free online resources) have or will affect computer science education.

Future studies should incorporate data on Computer Science Teacher Quality and evaluate the current class capacities to gauge the effectiveness of existing programs. While assessing teacher quality, a survey of the content and outcomes of computer science courses should be conducted to determine if disparities in quality match those in quantity across different demographics. Consequently, while assessing teachers and content, it will be essential to address potential barriers that may affect teaching not just computer science courses but other types of classes as well.

It is important to consider how state-wide and local educational policies influence computer science education. Funding data was not accessible for all years, so working with state agencies to provide transparency in funding equity across not just demographic groups but also boroughs to boost CS enrollment should be considered in future research. Furthermore, future studies should also delve into the effectiveness of these CS classes in steering students toward careers in computer science, thus assessing the practical impact of such educational initiatives. State agencies should survey to gauge how much influence their public school computer science course had in preparing their skills in this technology-driven economy.

While this study provides valuable insights into the evolving landscape of CS education, further comprehensive research is warranted to address the identified gaps and propel informed decisions and policies in advancing CS education inclusivity and effectiveness not just in New York City but across the country.

5. References

Arundel, K. (2023, November 1). *Schools increase offerings in computer science courses.* Industry Dive: K-12 Dive: Dive Brief. https://www.k12dive.com/news/schools-increase-computer-science-courses/698510/

Computer Science and Career and Technical Education Reports. (n.d.). Infohub.nyced.org. Retrieved December 12, 2023, from https://infohub.nyced.org/reports/government-reports/computer-science-and-cte-reports

Dvorkin, E. and Crowe, C. (2023, September). *Expanding on CS4ALL: Training NYC's Future Teachers to Integrate Computing Education.* Center for an Urban Future: Research. https://nycfuture.org/research/expanding-on-cs4all#:~:text=In%20an%20effort%20to%20close,science%20learning%20experience%20in%20each

Ezell, S. (2021, November 29). *Assessing the State of Digital Skills in the U.S. Economy.* Information Technology & Innovation Foundation. https://itif.org/publications/2021/11/29/assessing-state-digital-skills-us-economy/

Flórez, F. B., Casallas, R., Hernández, M., Reyes, A., Restrepo, S., & Danies, G. (2017). Changing a Generation's Way of Thinking: Teaching Computational Thinking Through Programming. *Review of Educational Research*, *87*(4), 834–860. http://www.jstor.org/stable/44667676

Google Inc. & Gallup Inc. (2016). Trends in the State of Computer Science in U.S. K-12 Schools. https://services.google.com/fh/files/misc/trends-in-the-state-of-computer-science-report.pdf

Hansen, M. & Zerbino, N. (2022, April 11). *Exploring the state of computer science education amid rapid policy expansion.* The Brookings Institute. https://www.brookings.edu/articles/exploring-the-state-of-computer-science-education-amid-rapid-policy-expansion/

Howell, S. (2023). *Technology Competition: A Battle for Brains*. Center for a New American Security. http://www.jstor.org/stable/resrep52276

LOCAL LAWS OF THE CITY OF NEW YORK FOR THE YEAR 2016 No. 177, Local Law No. 177 of 2016, Council Int. No. 1193-A of 2016 (2016). https://infohub.nyced.org/docs/default-source/default-document-library/locallaw177.pdf?sfvrsn=fbec43ee_2

New York City Department of Education. (2023). *Computer Science Education*. NYC Public Schools. https://www.schools.nyc.gov/school-life/activate-your-future/computer-science-education

New York State Education Department. (2023). *Frequently Asked Questions.* New York State Education Department: Knowledge, Skill, Opportunity: Standards and Instruction. https://www.nysed.gov/curriculum-instruction/csdf-frequently-asked-questions

Oldham, J. (2021). Computer Science for All? As a new subject spreads, debates flare about precisely what is taught, to whom, and for what purpose. Education Next, 21(4),44-51.https://www.educationnext.org/computer-science-for-all-as-new-subject-spreads-debates-flare/

Rasser, M., Lamberth, M., Riikonen, A., Guo, C., Horowitz, M., & Scharre, P. (2019). HARNESS AMERICA'S TALENT PIPELINE. In *The American AI Century: A Blueprint for Action* (pp. 13–19). Center for a New American Security. http://www.jstor.org/stable/resrep20445.9

Shashkevich, A. (2019, March 19). *Computer science college seniors in U.S. outperform peers in China, India and Russia, new research says.* Stanford University: News. https://news.stanford.edu/2019/03/19/comparing-skills-computer-science-undergraduates-internationally/#:~:text=Researchers%20found%20that%20the%20average,small%20and%20not%20statistically%20significant.

Slagg, A. (2022, June 23). *Teaching the Principles of Computer Science Early in K–12 Schools*. EdTech Focus on K-12. https://edtechmagazine.com/k12/article/2022/06/teaching-principles-computer-science-early-k-12-schools-perfcon

Smith, M. (2016, January 30). *Computer Science for All*. The White House: President Barack Obama. https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all

Stiffler, L. (2023, November 1). *New graduation requirements could get more girls taking computer science courses.* GeekWire, LLC. https://www.geekwire.com/2023/new-graduation-requirements-could-get-more-girls-taking-computer-science-courses/#:~:text=Eight%20states%20have%20taken%20that,met%20with%20computer%20science%20classes.

The New York City Department of Education. (2022). *Welcome to CS4All.* NYC Public Schools: Computer Science for All. https://sites.google.com/schools.nyc.gov/cs4allnyc/

U.S. Department of Education. (NCES 2021-021). *TIMSS 2019 U.S. Highlights Web Report.* Institute of Education Sciences, National Center for Education Statistics. https://nces.ed.gov/timss/results19/index.asp.

Vegas, E. & Fowler, B. (2020, August 4). *What do we know about the expansion of K-12 computer science education? A review of the evidence.* The Brookings Institution. https://www.brookings.edu/articles/what-do-we-know-about-the-expansion-of-k-12-computer-science-education/

6. Appendix

Please see HUDK4050_Final_Project_Analysis_(Katherine_Rojas_&_Maeghan_Sill.ipynb file for data analysis.

Source code for several visuals came from the Visualization.ipynb file shared in the HUDK4050 course at Teachers' College, Columbia University.