

Executive Summary

My friend is a Middle School math teacher. There are standard tests the school district gives out so the district can track the status of the various math classes. However, this data is not well utilized. The teachers do not get any use from this data. These teachers must take time out of their day to give students this test and they would appreciate some tangible benefit from it. I designed a program that will allow teachers to utilize this database of information.

My program allows teachers to select a portion of the database, sorted by teacher and/or class. They can then see statistics for the chosen sample. This will allow them to evaluate how a particular class is doing, or a particular teacher. It gives statistics for the chosen sample such as average and standard deviation, and for the database as a whole, to allow comparison. It also breaks down statistics by gender, so teachers can see any potential correlations. A teacher can access her class and see how that class compares to the school as a whole. This will help the teacher know if they need to improve or if they are exceeding the school average. This project will also create a histogram. Numbers are more valuable if seen visually.

In addition to viewing aggregate statistics, the teacher can pull out specific students that are struggling and those students that are above average. This information will allow teachers to tailor their class to specific students, allowing students to get more individualized teaching. It could also be useful to identify students that might be better placed in a higher or lower level math class, or at least receive additional tutoring. Another function, allows a teacher to pull up a particular student to see how they are doing compared to their class mates. This would be a nice starting point for determining how a student is doing. For example, if a student were struggling it would be useful to know if that student's poor scores are unique to that student or if they are ranked in the middle of the class. This would help a teacher understand if the problem is with the particular student or with an outside factor affecting the whole class.

In addition to these evaluation functions, this program also allows basic maintenance. There are user forms allowing teachers to search the database, edit entries in the database, add new students and enter test scores into existing students. These features are used to keep the database current.

Implementation

My project has six main functions. These functions are percentile ranking, searching the database, editing an entry, entering scores, adding a student, and generating statistics. Each of these functions can be accessed from a button on the ribbon. I will discuss the implementation of each of these functions in detail.

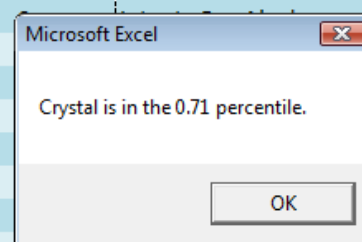
	A	B	C	D	E	F	G
1	ID	First Name	Last Name	Teacher	Grade Level	Class	School
2	37670	Rachel	Bowman	Barbara Hansen	7	Algebra 1	Mountain View
3	38272	Robert	Adams	Barbara Hansen	7	Algebra 1	Mountain View
4	32381	Skylar	Wagner	Barbara Hansen	7	Algebra 1	Mountain View
5	32209	Carson	Griffin	Barbara Hansen	7	Algebra 1	Mountain View

The database this project is working off has a row for each student. It includes basic information such as name, ID number, teacher, grade level, and class as well as the test scores. The functions within my project utilize this data to sort the results and show statistics for a specific portion of the data. For example, a teacher could choose to find the statistics for all Algebra 1 classes or only the Algebra 1 classes taught by her.

Percentile Ranking

The first function available in this project is percentile ranking. If you select a specific row in the database and click the ranking button in the ribbon, a message box will display telling you in which percentile that student ranks. This function compares the student's average score to the rest of the average scores in the database. This information will be useful to a teacher, as it tells them not only a student's grade, but also how that grade compares to the rest of the students in the school. A percentage grade doesn't always convey the right meaning if the class or school average is skewed. However, a percentile ranking will tell you a relative score.

1	36509	Mya	Ellis	Shelley Robinson	7	Algebra 1	Mountain View
2	39992	Miranda	Knight	Shelley Robinson	7	Algebra 1	Mountain View
3	36122	Patrick	Russell	Shelley Robinson	6	Intro to Pre-Algebra	Mountain View
4	24856	Stephen	Mills	Shelley Robinson			Mountain View
5	41368	Adam	Bell	Shelley Robinson			Mountain View
6	36423	Jeremy	Jordan	Shelley Robinson			Mountain View
7	29801	Reagan	Carpenter	Shelley Robinson			Mountain View
8	31779	Donovan	Montgomery	Shelley Robinson			Mountain View
9	33327	Sergio	Oliver	Shelley Robinson			Mountain View
10	35907	Leonardo	Chapman	Shelley Robinson			Mountain View
11	40637	Austin	Hill	Shelley Robinson	6	Intro to Pre-Algebra	Mountain View
12	29672	Aidan	Reed	Shelley Robinson	6	Intro to Pre-Algebra	Mountain View
13	40809	Brittany	Carr	Shelley Robinson	6	Intro to Pre-Algebra	Mountain View
14	31435	Kayla	Gilbert	Shelley Robinson	6	Intro to Pre-Algebra	Mountain View
15	25372	Crystal	Peters	Shelley Robinson	6	Intro to Pre-Algebra	Mountain View
16	30188	Cheyenne	Nichols	Shelley Robinson	6	Intro to Pre-Algebra	Mountain View
17	34746	Jordan	Gonzalez	Shelley Robinson	6	Intro to Pre-Algebra	Mountain View
18	37627	Jacqueline	Graham	Shelley Robinson	6	Intro to Pre-Algebra	Mountain View
19	27221	Nevaeh	Long	Shelley Robinson	6	Intro to Pre-Algebra	Mountain View



Searching the Database

The database of students is very large and hard to utilize, and this database is just a sample database containing only a fraction of the students a real database would. Because of this, I created a user form to search the database. This form allows the user to enter a student name (or a partial name), select a class and teacher from drop down lists, and search through the database.

The screenshot shows a spreadsheet-like table with columns: Name, Last Name, Teacher, Grade Level, Class, and School. A 'Find Student' dialog box is open, allowing search by Student First Name, Student Last Name, Class, and Teacher. A dropdown menu for the Teacher field is currently open, showing a list of names including Barbara Hansen, Frank Wilson, Kelly Williamson, Linda Tracy, Robert Samuels, and Shelley Robinson. The 'Find First' button is highlighted.

Name	Last Name	Teacher	Grade Level	Class	School
Bowman	Barbara Hansen		7	Algebra 1	Mountain View
Adams	Barba				tain View
Wagner	Barba				tain View
Griffin	Barba				tain View
Warren	Barba				tain View
Ramirez	Barba				tain View
Henry	Barba				tain View
Reyes	Barba				tain View
Taylor	Barba				tain View
Lowe	Barba				tain View
Lopez	Barba				tain View
Douglas	Barba				tain View
Mitchell	Barba				tain View
Gardner	Barba				tain View
Harrison	Barbara		7	Algebra 1	Mountain View
Black	Barbara		7	Algebra 1	Mountain View
Austin	Barbara Hansen		7	Algebra 1	Mountain View
Bryant	Barbara Hansen		7	Algebra 1	Mountain View
Duncan	Barbara Hansen		6	Intro to Pre-Algebra	Mountain View
Carlson	Barbara Hansen		6	Intro to Pre-Algebra	Mountain View

Clicking on the find first button will find the first row in the database that matches all specified search criteria. The information on the found record will display in the boxes next to the search fields. If this is not the entry the user was looking for they can click find next. This button will take them to the next row in the database also matching the search criteria specified. When there are no more entries in the database matching the search criteria, a message will pop up informing the user that no more matches were found.

This user form also allows the user to double click on a displayed result to copy the value to the search box. This would allow a user to change search criteria based on a prior search's results. Once the user has found the student they are looking for, they can close the form. The student they found will be highlighted ready to perform another function on that student record.

The 'Find Student' dialog box now shows search results. The 'Student First Name' field contains 'Shane' and the adjacent box also shows 'Shane'. The 'Student Last Name' field is empty, and the adjacent box shows 'Ray'. The 'Class' dropdown is set to 'Intro to Pre-Algebra' and the adjacent box also shows 'Intro to Pre-Algebra'. The 'Teacher' dropdown is empty, and the adjacent box shows 'Barbara Hansen'. The 'Find First' button is highlighted, and the 'Row #' field shows '22'.

Barbara Hansen	7	Algebra 1	Mountain View
Barbara Hansen	7	Algebra 1	Mountain View
Barbara Hansen	7	Algebra 1	Mountain View
Barbara Hansen	7	Algebra 1	Mountain View
Barbara Hansen	7	Algebra 1	Mountain View
Barbara Hansen	6	Intro to Pre-Algebra	Mountain View
Barbara Hansen	6	Intro to Pre-Algebra	Mountain View
Barbara Hansen	6	Intro to Pre-Algebra	Mountain View

Editing an Entry

Another basic function a user would need is an easy way to edit student entries. The edit form allows a user to update a student's name, grade, teacher, class, or school. In addition, it allows a teacher to add comments to that record. When a row in the database is highlighted, clicking the edit button allows the user to update that particular student's information. This works well with the search function, as the user can use that form to find a student. The row will be highlighted, so clicking the edit button will then allow the user to edit that entry.

When this form is opened, the selected student's information is displayed. The student ID number cannot be changed, but every other field allows the user to change the information. Name and grade level are text boxes, while teacher, class, and school are drop down lists. Once the user has changed the necessary information clicking update will close the form and save the changed in the database, overwriting the previous information.

The screenshot shows a web application interface. In the background, there is a table with student information. The table has columns for Name, Last Name, Teacher, Class, School, and a score column. The table is populated with student names like Bowman, Adams, Wagner, Griffin, Warren, Ramirez, Henry, Reyes, Taylor, Lowe, Lopez, Douglas, Mitchell, Gardner, Harrison, Black, Austin, Bryant, Duncan, and Carlson. Overlaid on this table is a modal window titled "Edit Student Data". This form contains fields for Student ID Number (32037), First Name (Shane), Last Name (Ray), and Grade Level (6). It also has dropdown menus for Teacher (Barbara Hansen), Class (Intro to Pre-Algebra), and School (Mountain View). There is a text area for Teacher Comments with the placeholder text "Teacher comments are typed in this box". An "Update" button is located at the bottom right of the form. The "Class" dropdown menu is open, showing options: Algebra 1, Intro to Pre-Algebra, and Pre-Algebra.

Name	Last Name	Teacher	Class	School	Test 1	Test 2
Bowman					62	
Adams					55	
Wagner					64	
Griffin					67	
Warren					61	
Ramirez					67	
Henry					72	
Reyes					74	
Taylor					72	
Lowe					68	
Lopez					74	
Douglas					73	
Mitchell					76	
Gardner					66	
Harrison					84	
Black					82	
Austin	Barbara Hansen	7	Algebra 1	Mountain View	78	
Bryant	Barbara Hansen	7	Algebra 1	Mountain View	89	
Duncan	Barbara Hansen	6	Intro to Pre-Algebra	Mountain View	55	
Carlson	Barbara Hansen	6	Intro to Pre-Algebra	Mountain View	61	

Entering Scores

As the main purpose of this project is to aggregate students' math scores, it is important for the user to have an easy way to enter test scores into the database. As tests are administered and scored by individual teachers (as opposed to with a scantron or other technology), the easiest way to do this is through a user form.

The form to insert scores displays the information for the active row along the top of the form. This information cannot be changed; it is there so the user knows which record they are working with, and can verify that they are entering the scores into the correct record. There are four boxes to enter in test scores, one for each test given in the year. Once the scores are entered, the user can click next. This

will save the scores entered and move to the next row. Only boxes with a number entered in will be updated. If the user leaves a box blank, the corresponding cell on the sheet will not be changed. Once the user is finished entering scores clicking close will exit the user form.

The screenshot shows a spreadsheet with columns labeled D (Teacher), E (Grade Level), F (Class), and G (School). The data rows show Barbara Hansen as the teacher for Grade 7 Algebra 1 at Mountain View. An 'Insert Scores' form is overlaid on the spreadsheet. The form contains fields for ID Number (32037), First Name (Shane), Last Name (Ray), Class (Intro to Pre-), and Teacher (Barbara Hansen). Below these fields are four test score input areas: Test 1, Test 2, Test 3, and Test 4, each with a text box and 'out of 100' label. To the right of the test input areas are 'Next' and 'Close' buttons. The spreadsheet rows are partially visible behind the form, showing the same teacher and class information.

The database is sorted first by teacher and then by class. This makes it easy for a teacher to enter class scores into the spreadsheet. The teacher will need to find the first occurrence of their specific class to start the form on. Then when they click next, it will take them through all of their students in alphabetical order by class. One way for the teacher to easily find a good spot to start entering scores is the search function. By searching for the first student on their roles, it will make that student the active row. The rest of the class will be in alphabetical order after that student, making it easy to use the next button on the form.

Adding a Student

Schools often have high turnover. Therefore, I created a form to make entering new students into the database quick and easy. Instead of the user scrolling down the page searching for the bottom of the database, they can click a button, enter the data into the form and be done. This form is very similar to the edit form; however, it has more information on it. There is some information like, gender and ID number that the user couldn't edit before, but the user will want to add them for a new student. This is a very simple form. The user enters the requested information, choosing teacher, class, and school from a drop down list. When all information is entered, the user clicks save. This creates a new row in the database with the entered information.

	D	E	F	G
	Teacher	Grade Level	Class	School
	Barbara Hansen	7	Algebra 1	Mountain View
B	Add Student			
B	ID Number	First Name	Last Name	Gender Grade
B	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/> <input type="text"/>
B	Teacher	Class	School	
B	<input type="text"/>	<input type="text"/>	<input type="text"/>	
B	Teacher Comments			
B	<input type="text"/>			
B				Save
B				
B	Barbara Hansen	7	Algebra 1	Mountain View
B	Barbara Hansen	7	Algebra 1	Mountain View
B	Barbara Hansen	7	Algebra 1	Mountain View

Generating Statistics

Now that the maintenance functions of this project are done, the important part of this project is the statistics. Teachers or other school district employees want to be able to pull out information about a group of students. The statistics form allows the user to choose a combination of class and teacher. Then the user must select which set of data they want to analyze; an individual test, or the overall score. Once this information is chosen from the drop down lists, the user has two options. These options are show chart and statistics. The statistics form is accessed by clicking the Stats button in the ribbon. In addition, two statistical functions are accessed from the ribbon. These functions, called struggling and above average, pull out students that are in the top or bottom deciles.

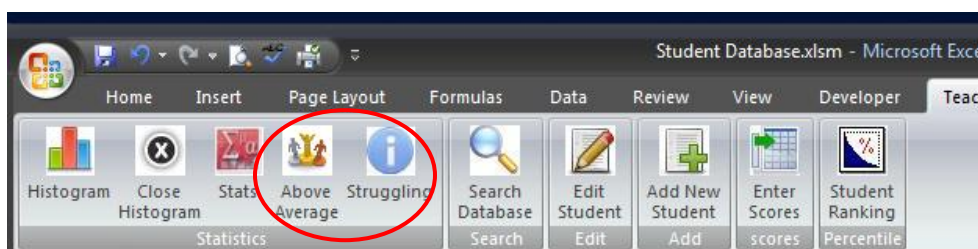
Aggregate Student Data

Choose Class
Algebra 1

Choose Teacher

Choose Test
Overall

Show Chart
Statistics

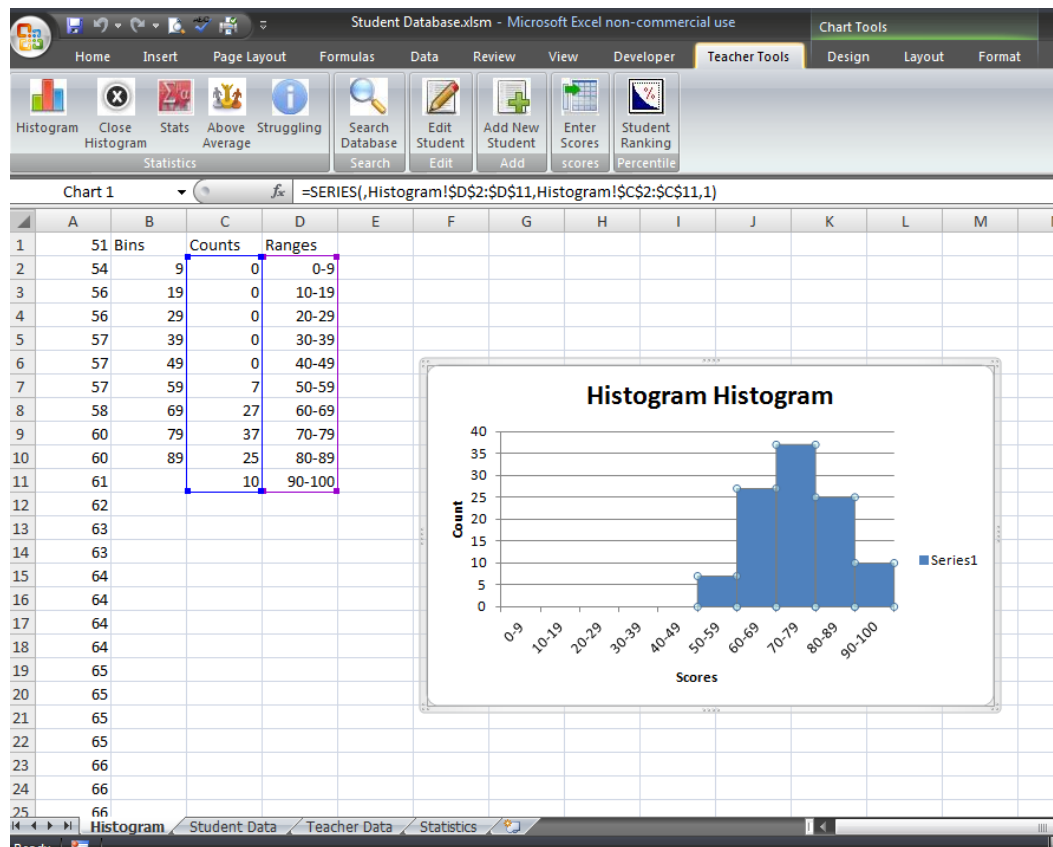


Show Chart

The show chart button creates a histogram of the selected data, and shows a chart displaying that histogram. The applicable test scores are pulled from the database and put in a new sheet. They are then sorted and grouped in bins. A chart is created from the organized data and displayed on a new worksheet. This allows the user to visually see how the selected group of students is doing.

The histogram should approximate a bell shaped curve. If the curve is skewed to the left, that indicates a problem. Scores are lower than the statistical probability. This lets the teacher know that there is a problem and that they need to do more research to find out what that problem is. If the curve is skewed right, this shows that more students are above average than normal. This isn't necessarily a problem, but it would still be useful to find out what is causing that skew. If the reason can be discovered, they could potentially copy it and apply that to other classes.

Every time the histogram button is clicked, the macro will delete the existing histogram without a warning and show the new histogram. This means it is not necessary for the user to delete the sheet before running a new sample through the macro. However, I have provided a button on the ribbon for the user to delete the histogram if they desire.



Statistics

Clicking the statistics button takes the user to a new worksheet set up to display statistical information. At the top of the page the class and teacher are displayed as well as the data the statistics are coming from, to remind the user which students they are looking at. Next is a box containing mean, median, mode, and standard deviation for the total group, and broken up into male and female. All scores matching the criteria entered in the drop down boxes on the form are taken from the student database. They are put into a new worksheet where the calculations are done and stored in variables. Then the new worksheet is deleted. This allows the user to ignore the calculation and not have to deal with a long list of numbers. In fact, the user never sees this happening at all.

Statistics

Class: This sample is **lower** than average database scores.

Teacher:

Test:

	Total	Female	Male
Mean	75.8	76.9	74.4
Median	75.0	75.0	75.0
Mode	74.0	74.0	76.0
Std Dev	8.6	9.3	7.5

Compare to:

	Total	Female	Male
Mean	76.0	76.9	75.3
Median	75.0	74.0	74.0
Mode	75.0	75.0	77.0
Std Dev	9.3	9.4	8.2

Total:	<input type="text" value="98"/>
Male:	<input type="text" value="43"/>
Female:	<input type="text" value="55"/>

Total:	<input type="text" value="401"/>
Male:	<input type="text" value="201"/>
Female:	<input type="text" value="200"/>

Student Data / Teacher Data / Histogram / **Statistics** / Decile

The statistics are given for both the sample the user searched and the database as a whole. This is given as a comparison. The user would need to know what the average overall is to know if this class average is good or bad. The page also analyzes the differences between the sample statistics and database statistics and returns a message on the worksheet if this sample is higher or lower than the database average.

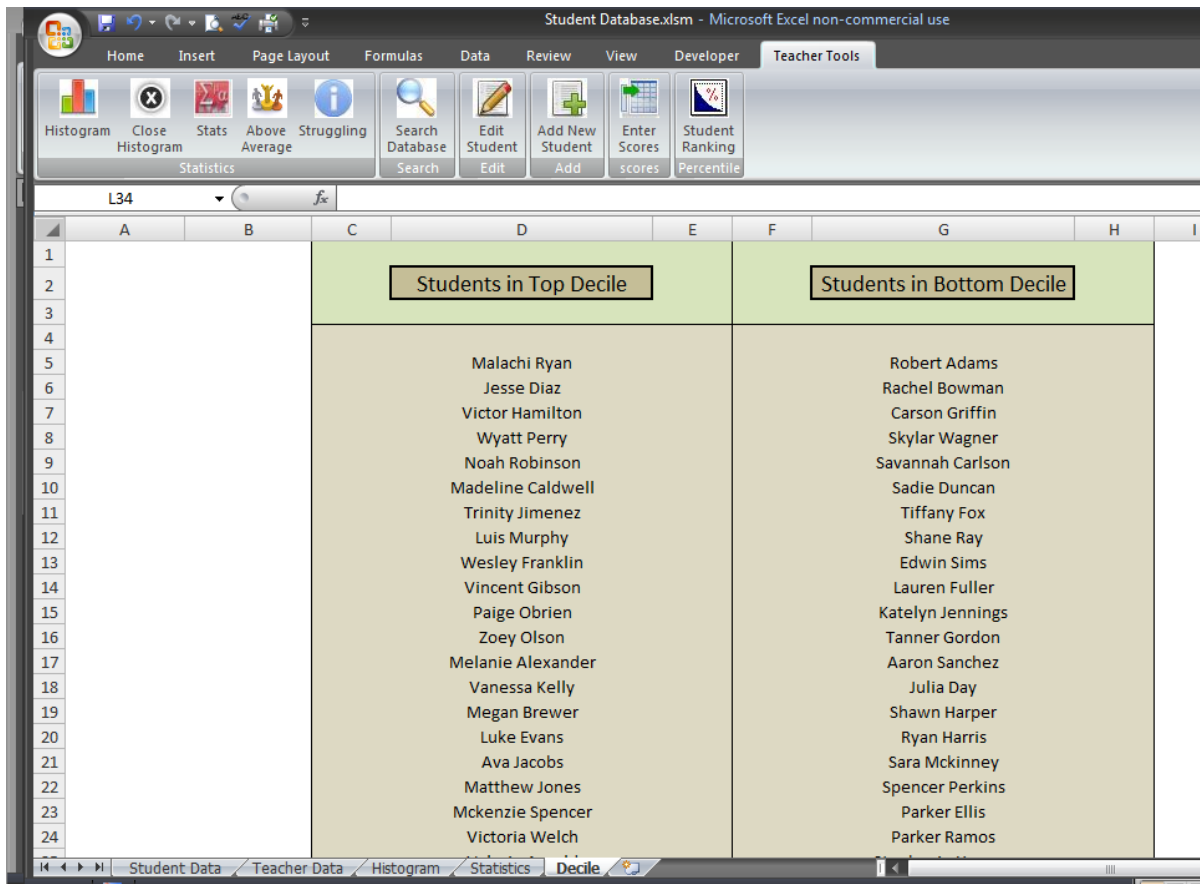
I created this macro to display common statistics that are used to analyze grades. Individual teachers utilize these statistics in different ways. They may want to know the mean and standard deviation to create a grade distribution. In addition comparing mean and median will give insight on possible outliers.

Struggling Students

Clicking the button labeled struggling pulls out a list of students who rank in the bottom decile and displays them for the user to see. This macro analyzes every row in the database and finds that student's ranking based on his or her overall score. If that ranking is in the bottom ten percent the name is added to a list on a sheet made to capture the names. This macro is useful to a teacher so they can see not just who is failing, but those students that rank significantly below the school average. These are students who could use extra help and tutoring.

Above Average

The button for above average works identically to that for struggling students. However, this pulls out those students in the top ten percent of the school. This information can be used to identify students that are probably bored and could be moved up or given extra assignments to keep them engaged. Again, these students' names are displayed on the same sheet as the struggling students.



	A	B	C	D	E	F	G	H	I
1									
2				Students in Top Decile			Students in Bottom Decile		
3									
4									
5				Malachi Ryan			Robert Adams		
6				Jesse Diaz			Rachel Bowman		
7				Victor Hamilton			Carson Griffin		
8				Wyatt Perry			Skylar Wagner		
9				Noah Robinson			Savannah Carlson		
10				Madeline Caldwell			Sadie Duncan		
11				Trinity Jimenez			Tiffany Fox		
12				Luis Murphy			Shane Ray		
13				Wesley Franklin			Edwin Sims		
14				Vincent Gibson			Lauren Fuller		
15				Paige Obrien			Katelyn Jennings		
16				Zoe Olson			Tanner Gordon		
17				Melanie Alexander			Aaron Sanchez		
18				Vanessa Kelly			Julia Day		
19				Megan Brewer			Shawn Harper		
20				Luke Evans			Ryan Harris		
21				Ava Jacobs			Sara McKinney		
22				Matthew Jones			Spencer Perkins		
23				Mckenzie Spencer			Parker Ellis		
24				Victoria Welch			Parker Ramos		

Learning and Conceptual Difficulties

The most difficulty I encountered in this project was with the histogram. To begin with I tried to record myself using the built in histogram generator. However, I could not get the histogram to generate based on a variable range of data. The data used to create the histogram will be a different size every time the macro runs. I could not find any good resources on making the built in function work, so I instead searched for other people's solution to this problem. I eventually found a sub procedure that would accept an array containing the data and a specified number of bins and create a histogram. Once I had a histogram it would be easy to build a chart from that histogram. However, I couldn't get this sub procedure to work. I got all of my data into an array, but I got an error every time I tried to pass the array down to the histogram sub procedure. I then went back online and found another solution. This one wasn't as nice as it had to read the data off a sheet. It would create a histogram and corresponding chart based on the selected range. This wasn't too hard to set up however. I just made a loop to pull out the data matching the specified criteria and copied it to a new sheet where the histogram was created.

Another issue I had in this project was with pulling out only the specified data. The user could select a teacher and a class and analyze only that specific data. I thought this was working fine until I noticed some of the samples were much larger than they should be. I realized two of my classes had the word "Algebra" in them. I was looping through the database pulling out records using the `instr()` function. However, that function would bring back both classes because they both had Algebra in them. It took me a while to find out how to identify rows that matched multiple criteria without using the `instr()` function.

A major problem I ran into was with creating the ribbon. I used the ribbon wizard we used in class. However it didn't work. Not only did it not work, but it crashed my project. I was very worried that it wouldn't be able to recover my data. Even though I had saved frequently, it gave an error saying the data was corrupted by the ribbon wizard. However, almost all the data was recovered. I spent some time working with the ribbon wizard trying to discover what I had done wrong (using a new workbook, not my project again), but I never could get it to work. I finally used the alternate way of creating ribbons. This was harder and I had to spend more time understanding how that worked.

This project also helped me learn more about utilizing built in excel functions. I needed to be able to find the average of a range and then display that average on the worksheet without keeping the range of data. I learned more about the difference between using the formula method versus `application.worksheetfunction`. I also learned more about using ranges. The ranges I was using the functions on were variable size. I had to find a way to write the function so that it would adapt to the size. I became more comfortable with concatenating on variables within a loop to identify a range. One examples of a range I used was, `Range(column & "2:" & column & lastRow)`.

I also got a lot of practice using arrays. I pulled out the scores for the females and the scores for the males and put them in separate arrays. Then I ran functions on them to bring back the statistics. This got complicated because I had so many separate sets of data I was working with. I had three groups for the sample and three overall. I had to label my variables well so I could remember what I was doing.

As I was building this project I thought of one function I would like to add, but couldn't get to work. I built a macro that would filter the database and show only students matching the class and teacher selected. This would allow a teacher to see only his or her students. This was not too difficult. The problem was using the filtered results in another macro, such as the insert scores macro. It would be easier for a teacher to insert scores for one class if the only visible rows were the students for that class. However when the macro filters the database, the records keep their original row number. They are shown as one continuous range, but the actual rows are from multiple spots in the workbook. Therefore, it is hard to use a loop to work across the range. I needed a way to record the actual row numbers of the filtered selection. I considered using a loop similar to the one I used in the search user form; however, I felt it was not worth it. I instead sorted the database on teacher and class. The teacher just has to search for the first student in his or her class, and the class will be in a continuous range. More rows will be visible before and after the desired range, but I felt this was not a major problem.

Assistance

I did not receive any assistance from others working on this project. I did get some code from online sources to help me create the histogram. I adapted it to fit my project, but the base was not mine.