EDAV Project 1: The Class

For the first project, we did not have one overarching question we wanted to answer so we instead created analyses and visualizations answering multiple questions that we found interesting. These can broadly be broken up into student experience, gender, and relationships between answers.

Regarding experience, we first created a few graphs comparing overall experience of students, including their experience level, knowledge of different tools, and preferred text editor.

Then we focused on how differences in gender affected students' responses. Before we started with the gender-specific visualizations, we created decision tree to predict the gender of students who did not indicate whether they preferred male or female pronouns (one student answered that they had no preference, and another student did not answer).

Finally, we created several graphs to evaluate whether there was a relationship between different survey answers, such as students' overall experience level and the number of tools they know.

This report proceeds with a brief description of preliminary work we did on the data set, followed by the analyses and visualizations.

I. Preliminary Work and Data Cleaning

Our first step involved loading and cleaning the data so it would be easier to work with. To clean the data, we first wanted to delete all the blank columns. We built a function that would identify the names of the blank columns and then deleted those columns from the final table.

Next, we renamed the remaining columns with simpler, shorter titles than their original names.

Then we made sure the survey responses that had been typed were uniformly worded. We identified the different terms people used for their school program, and converted unique answers to be consistent with the other programs. We did the same for the text editor used.

We also created dummy variables for both the tools people knew and the text editor people used. The vast majority of people had either multiple tools or multiple text editors listed, so to simplify we added new columns for each potential Tool or Text Editor. For each student, there would be "1" in the box if the student listed that Tool or Text Editor in his response or a "0" if he did not list it. This allowed us to more easily do visualizations analyzing the number of students who know certain tools.

As part of this preliminary work, we also converted each experience level ("None", "A little", "Confident", and "Expert") into numerical representations and added a column that calculated the average experience level for each student.

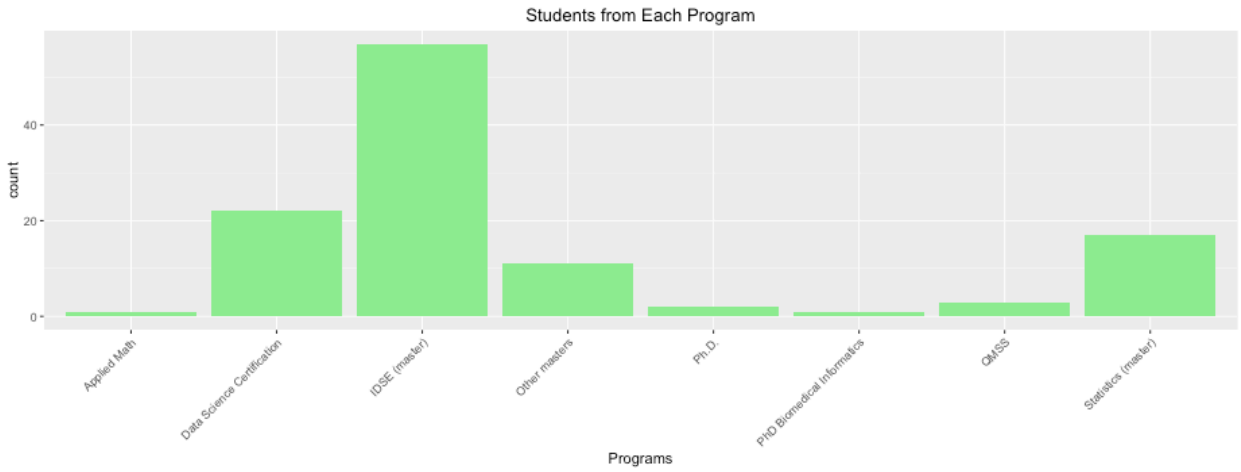
II. Analysis and Visualization

For our analysis and visualization, we decided to focus on several questions that interested us. The first questions dealt with basic questions about students' experience with different tools and text editors. We then decided to make more detailed graphs examining how students' answers differed based on their gender and program. Finally, we created graphs trying to identify correlations among different tools and experience levels.

A. Text Editors and Tools

1. Number of Students from Each Program

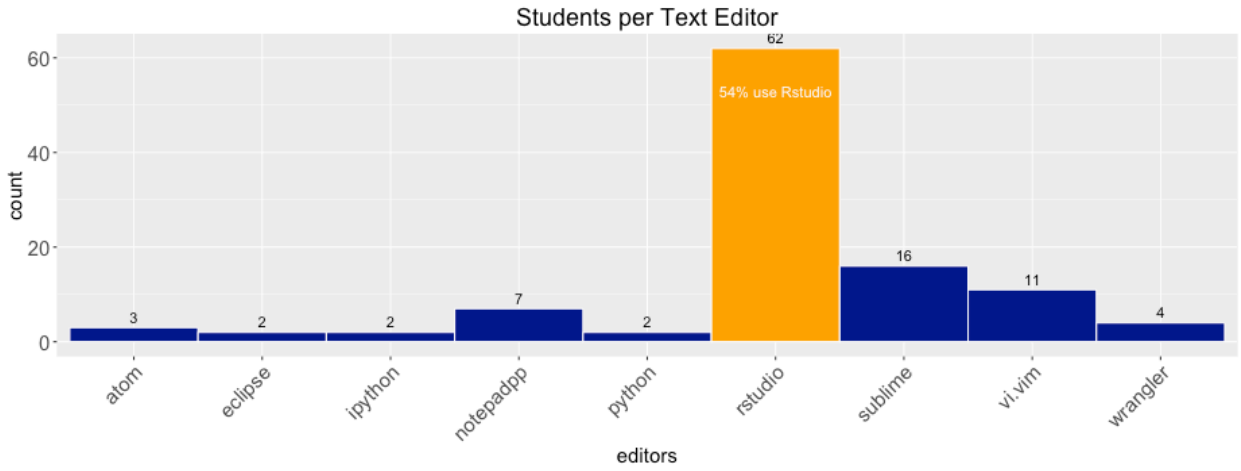
Before we dive into some of the more detailed graphs, below is a graph simply illustrating the number of students from each program.



Unsurprisingly, the majority of students in this course are in the Data Science Master's or Data Scene Certification program. Since this is a required course for both programs, this program breakdown is as we would expect.

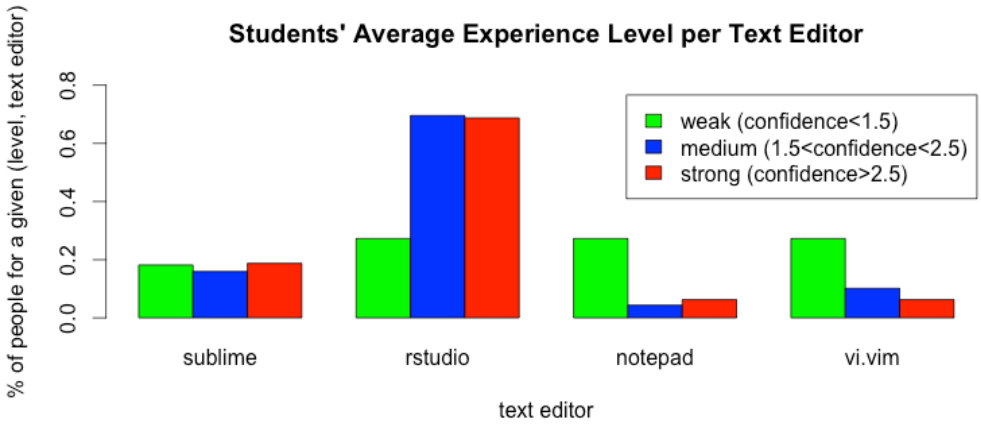
2. Text Editor Use

The following is a basic plot showing the distribution of the text editors used by students. Clearly most students prefer Rstudio, which we believe means that R is probably the language they use most often since Rstudio can only be used to write R scripts. To prevent the graph from becoming too cluttered, we only included text editors that were chosen by at least two students.



3. Text Editor and Average Experience

Next, we explored the average skill level of students as a function of their preferred text editor (using the four main test editors, notepad, sublime, Rstudio, vim). We divided the average confidence level of students into three categories, weak (average confidence less than 1.5), medium (average confidence between 1.5 and 2.5), and strong (average confidence between 2.5 and 4).

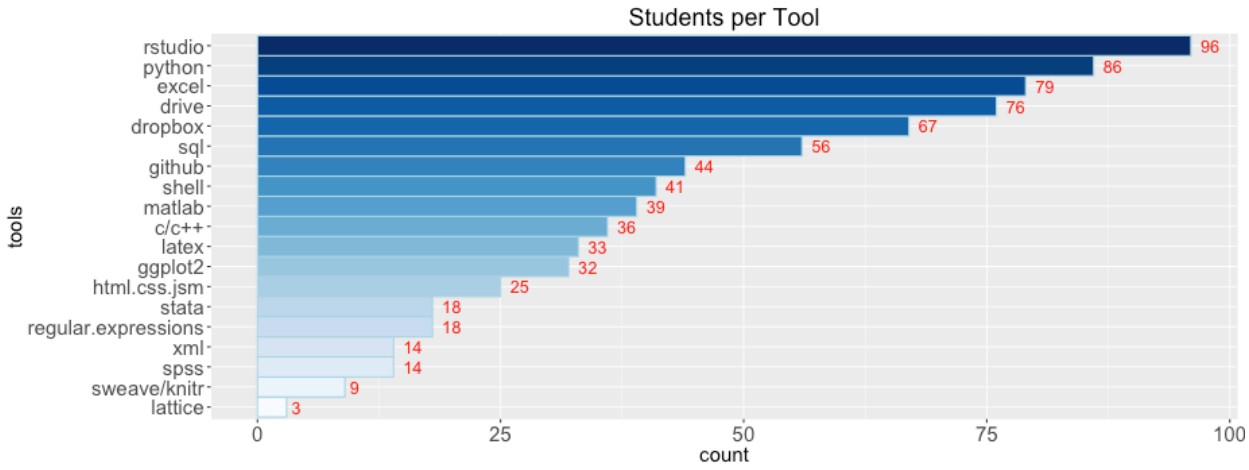


We would normally expect users using vi.vim to represent the "strong" cluster but in fact most of the people in the strong and medium cluster prefer RStudio. To understand that, we need to remind the readers how the average confidence is computed. The average confidence is the mean of the skills in:
-R data manipulation & modeling
-R graphics basics
-R advanced
-R markdown
-Matlab
-Github

Thus, since the average confidence is mainly a function of skills in R, it is normal to see the students with the highest average experience chose RStudio as their text editor.

4. Students per Tool

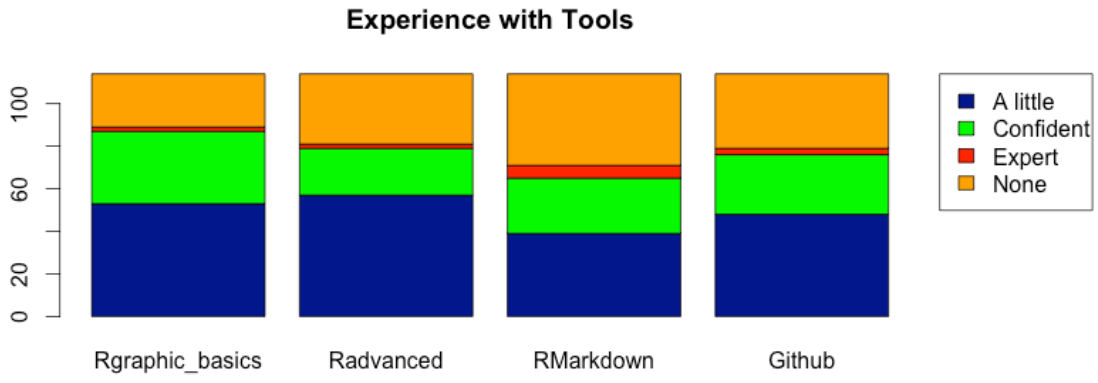
We then created a graph showing the number of students who have experience with each tool listed in the survey.



R and Python are by far the two most popular languages used by students. Since, as discussed above, the majority of the class is made up of students in the data science programs, it makes sense that RStudio is the most popular text editor.

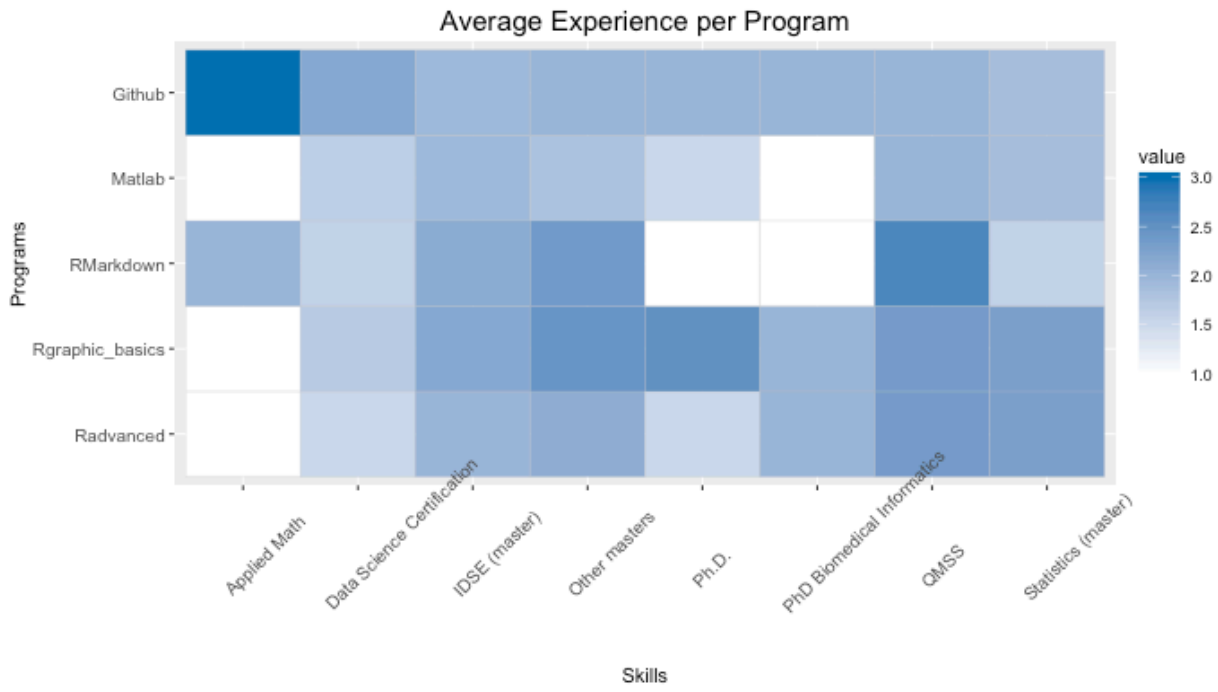
5. Experience Level with Tools

This graph shows the distribution of experience for four different tools, Basic R Graphics, Advanced R, RMarkdown, and Github. As we can see, very few students are experts in any of these tools. Hence, most teams will probably not have an expert in each of these tools.



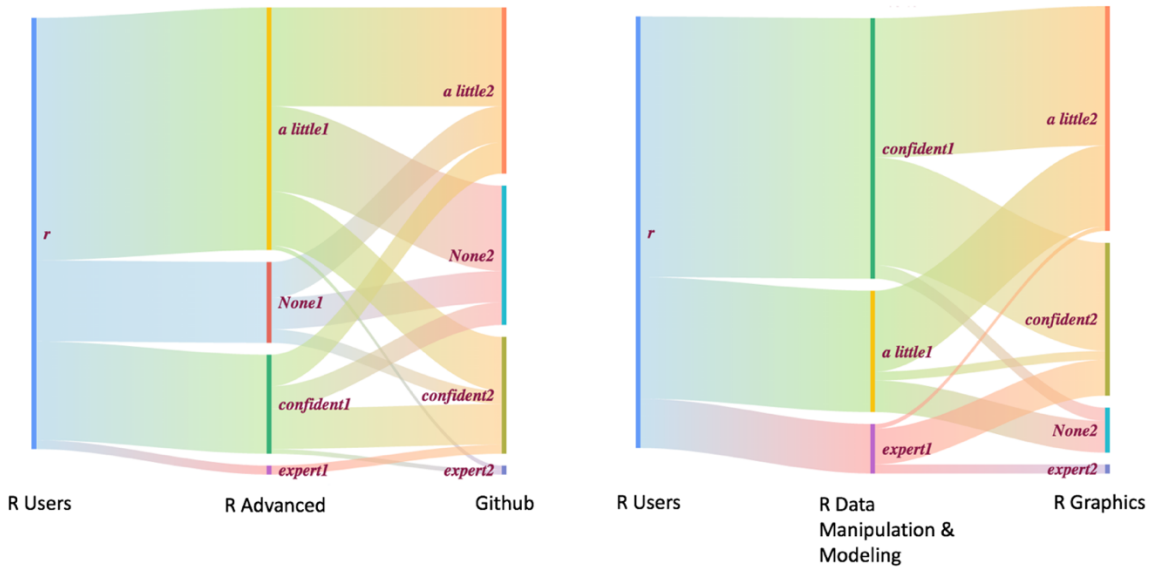
6. Average Experience in Tools per Program

Next we created a heat map that showed the average experience in five different tools for students from each program. With a few exceptions, students from most programs have an average experience of 1.5 to 2.5 in each of the tools listed.



7. Experience Level per Tool

The following pair of graphs are Sankey diagrams illustrating the proportion of students per experience level for each of the tools listed on the x axis.

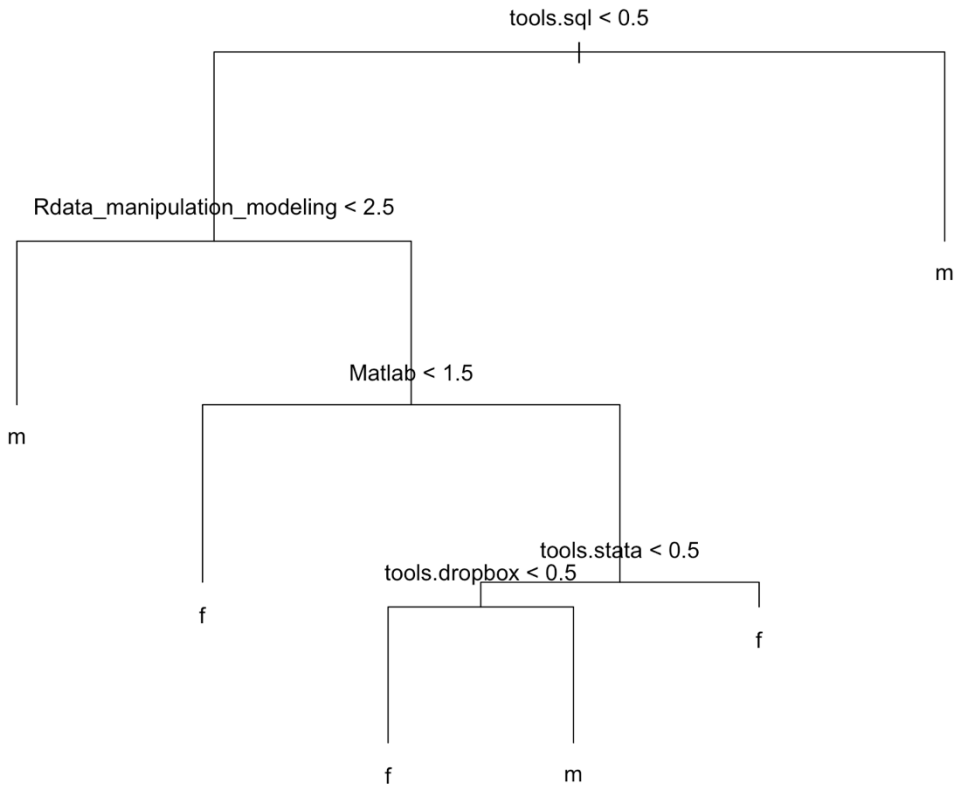


We can see from the first Sankey diagram (on the left) that on average, whatever the level in “R advanced” a given student has, he will have the same odds to be a confident, a medium or a beginner Github user. The main takeaway from this observation is that for the majority of the people in the class, their Github skills do not depend on their R skills. It is logical since Github is a tool designed mostly for software engineers than for statisticians. Thus, for most of the people, one of the challenges of this class will be to learn how to master Github.

The second Sankey diagram clearly shows that on average a person who has a skill level *n* in “R data manipulation & modeling” will have a skill level *n-1* in R graphics. Consequently, people tend have a lower level in R data visualization than in R data manipulation. This last point proves that this class is truly relevant because the majority of the students need to improve their data visualization skills.

B. Gender

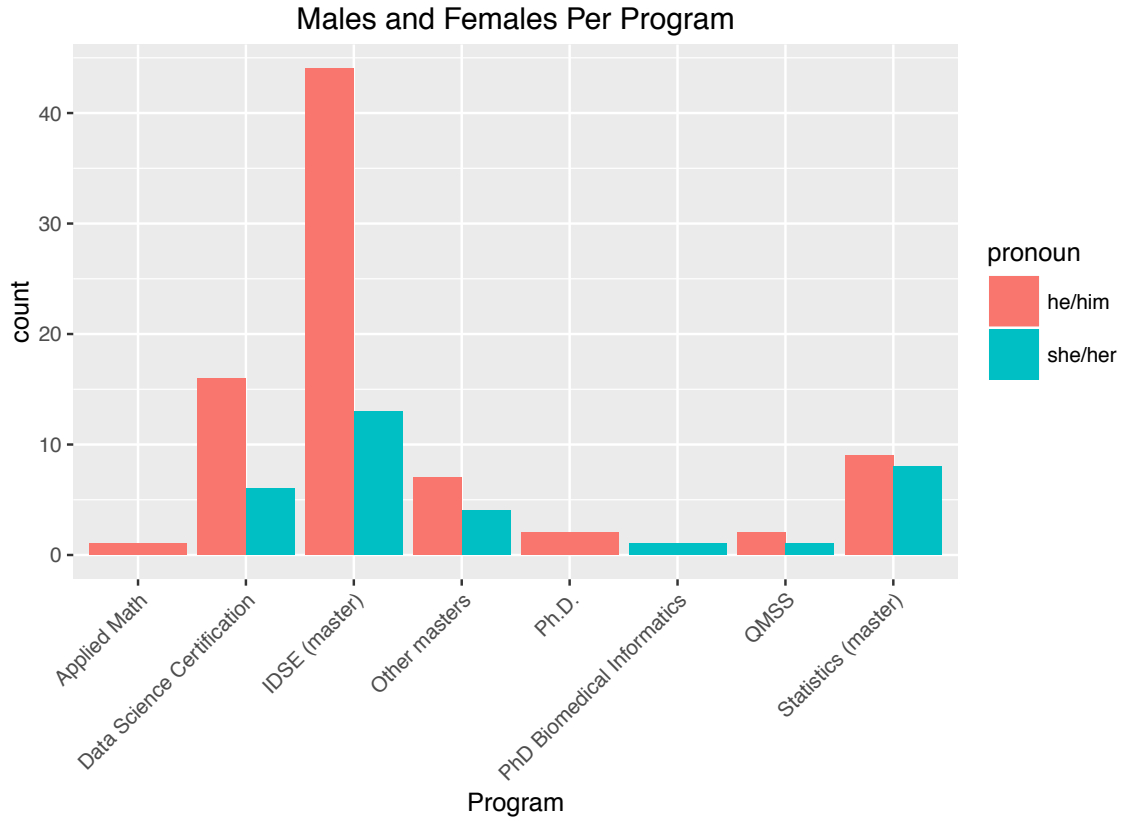
The next few graphs examine the differences in responses between men and women. First, however, we tried to predict the gender of students who did not choose a preferred gender. In the survey responses, there was one student who chose N/A for their preferred pronoun and one student who said they had no preferred pronoun. To try to keep the graphs simpler, we built a decision tree to try to predict whether the "N/A" and "Doesn't matter" responses were from a male or female.



The explanation of how the decision tree was created is in the “Decision Tree Explanation” file in this project folder.

1. Number of Males and Females from Each Program

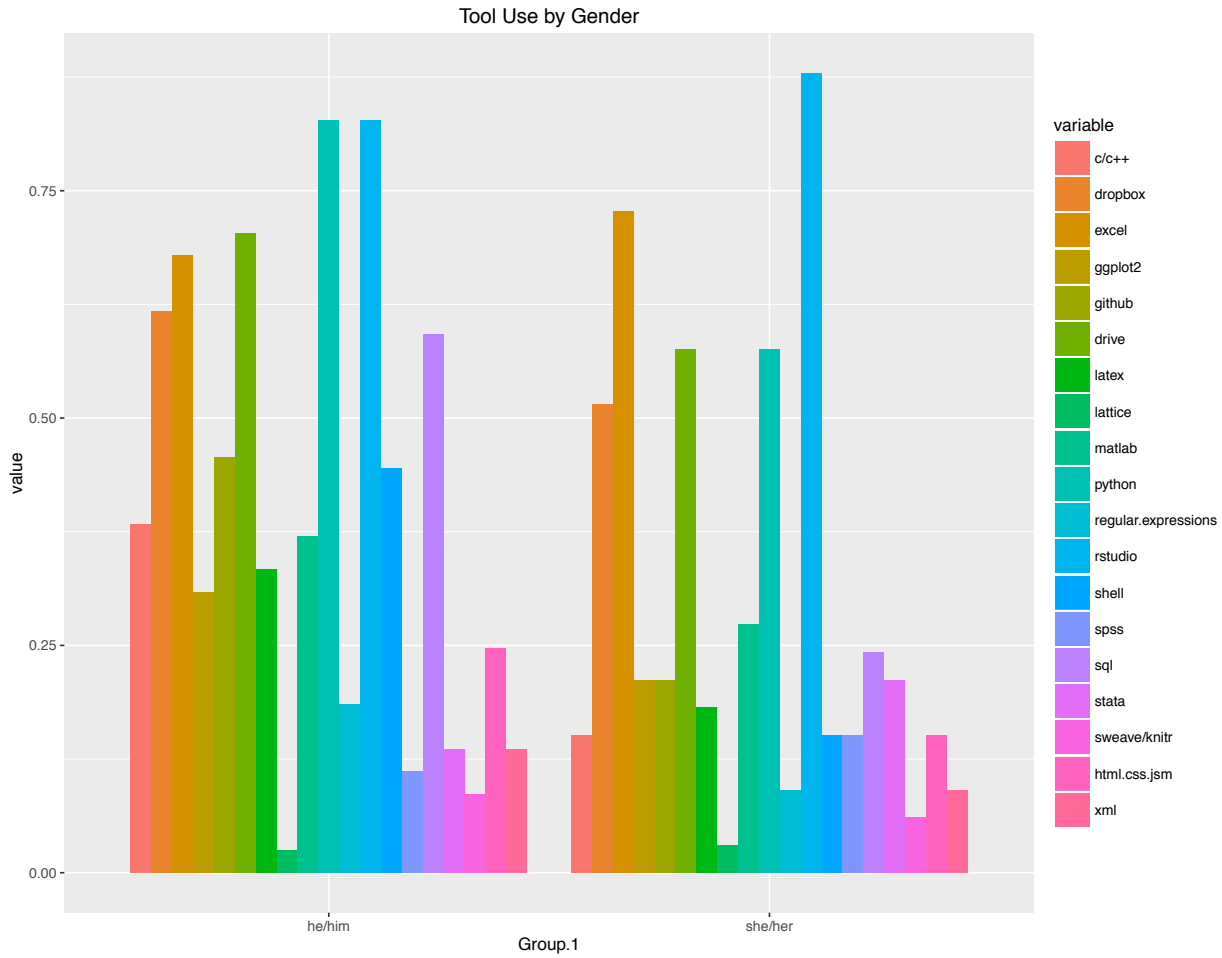
The first graph in this section illustrates the number of males and females from each different program.



Overall, the proportion of males to females skews highly male except for PhD students and the Statistics master's students. Columbia University in general has a nearly even split of men and women, with women making up 47% of all students in Columbia's graduate and professional schools. Thus, the gender makeup of men to women in this course is not consistent with Columbia in general (http://www.columbia.edu/cu/opir/abstract/opir_enrollment_gender_1.htm).

2. Tools vs. Gender

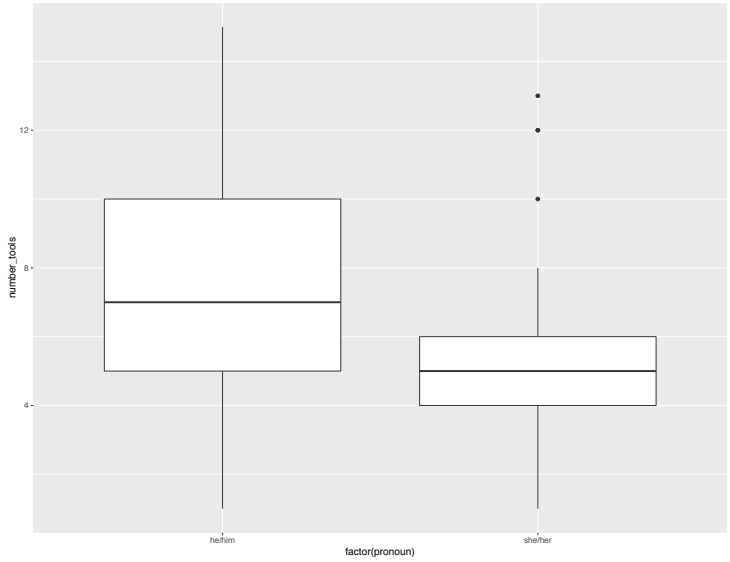
Our next graphs focus on the differences in tools according to gender. The first is a bar graph that shows the proportion of students who have experience with each tool according to their preferred pronoun. There are 19 tools available. Since there is an uneven number of men and women in the class, we had to normalize the values by dividing by the number of men and women in the class (80 and 32, respectively).



When the counts are normalized, we can see that the overall shapes of the distributions of tools are very similar between men & women. However, significantly more men than women use C/C++, Shell, and SQL, which are often used by software developers, DBAs, etc. This may be due to more men than women working in those fields, at least in the U.S. (http://www.techrepublic.com/blog/oftware-engineer/it-gender-gap-where-are-the-female-programmers/).

3. Number of Tools per Gender

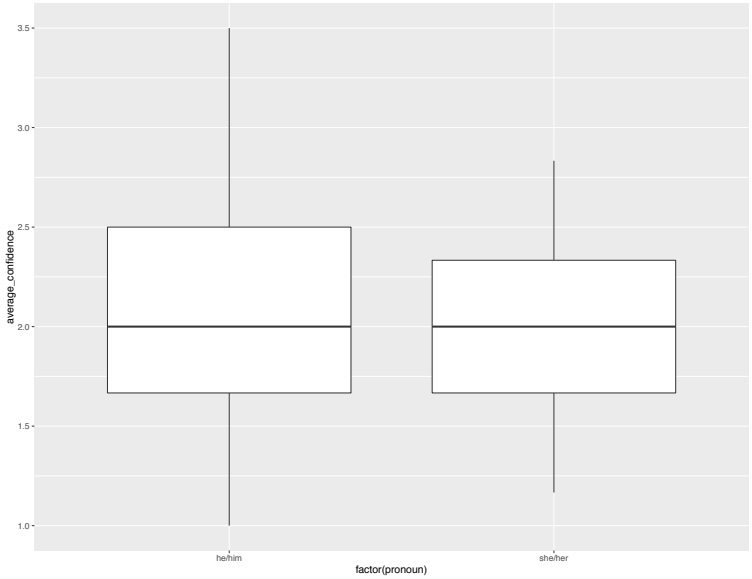
The following is a box plot of the number of tools known, broken down by gender.



This box plot shows that women overall report knowing fewer tools than males.

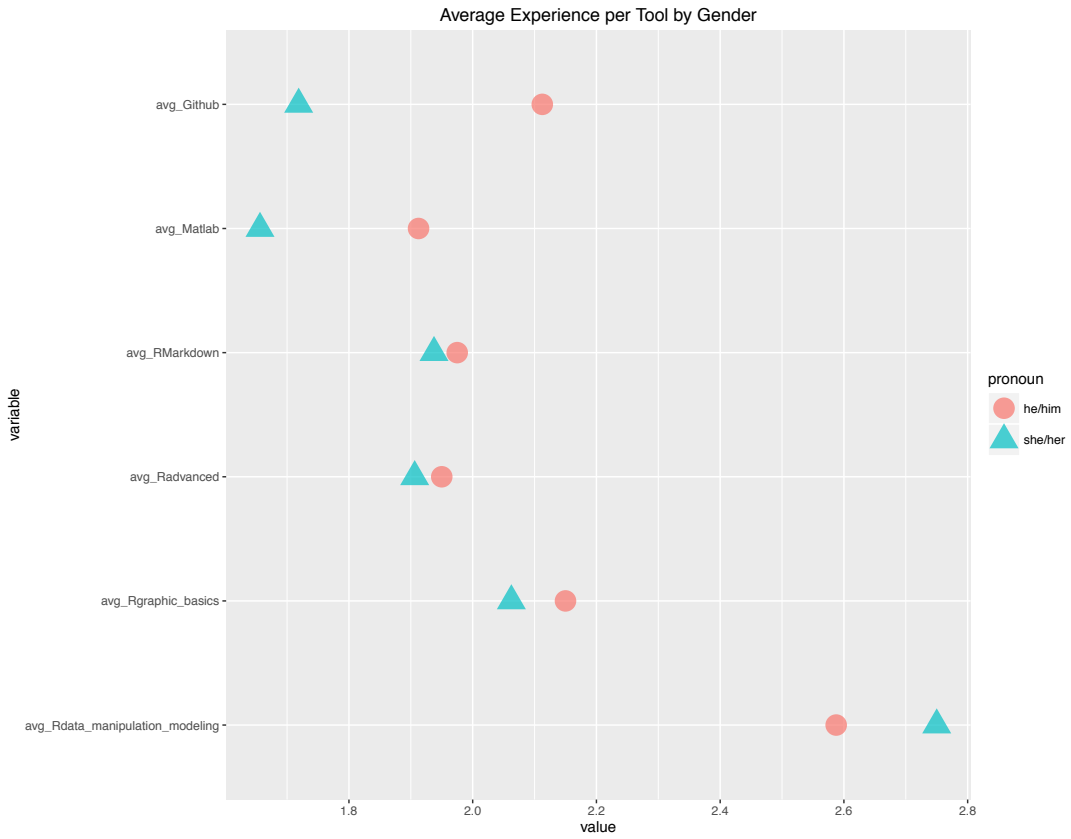
4. Experience Level vs. Gender

The next set of graphs also looks at gender, but instead of focusing on the quantity of tools known by students, focus on their level of expertise in the tools they know. Unlike the box plot above, the values for this box plot are nearly even between men and women.



5. Average Experience per Tool by Gender

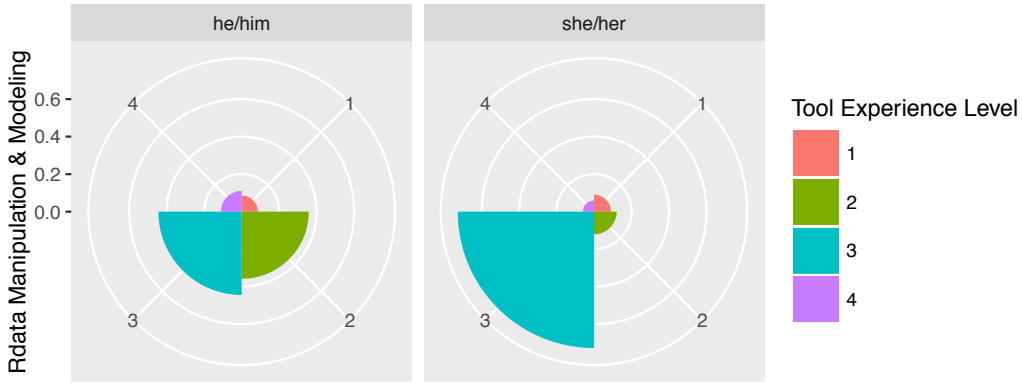
The following graph plots the average experience of each gender for six tools (Github, Matlab, R-Markdown, R advanced, R graphic basics and R data). Except for R Data Manipulation and modeling, males have a higher average experience for each of the tools listed.

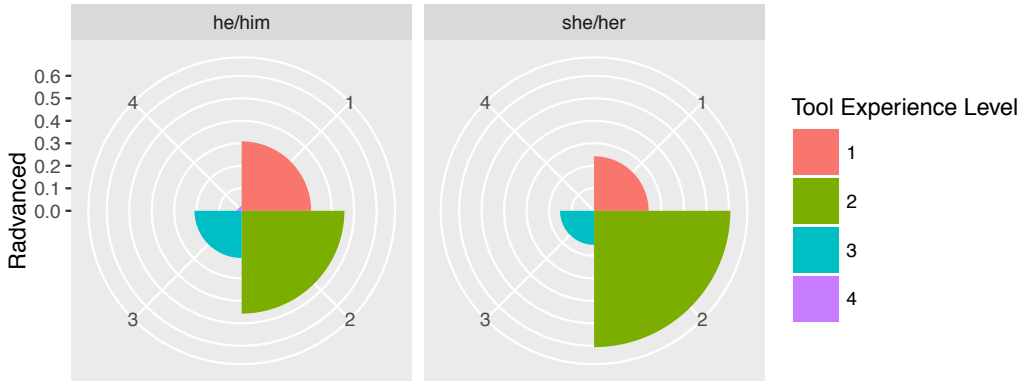


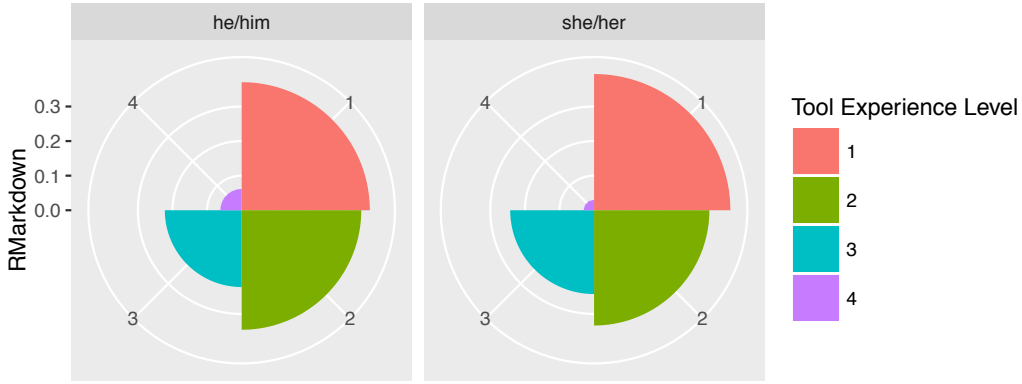
We can conclude from this graph that, generally speaking, female students tend to be slightly more modest about evaluating their experience level for most of the programming and analytical tools.

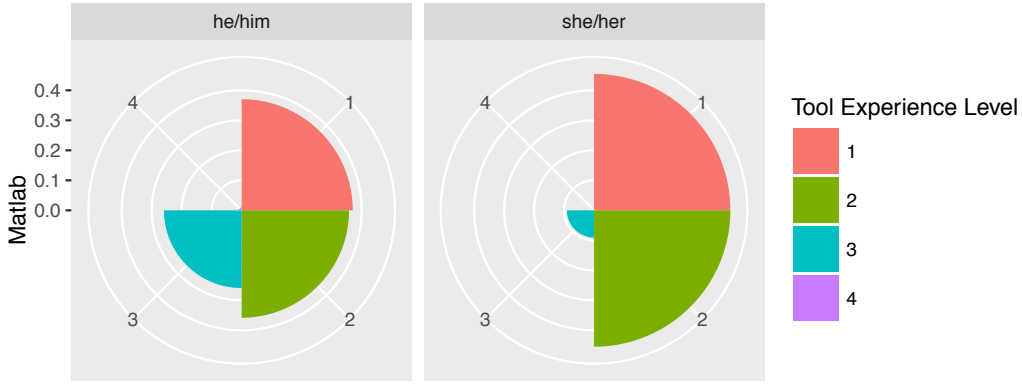
6. Breakdown of Specific Tools Experience by Gender

Below is the breakdown of experience levels for each gender for the six tools listed above.









Based on the shapes of the graphs we can find that the proportions follow quite similar pattern for both genders; the choice of level 2 dominates all programming and analytical tools besides

Rdata Manipulation & Modeling, where most people were at level 3. This can be explained by this bring the Spring semester. Students are at least in their second semester in their program, and gained experience with Rdata Manipulation and Modeling over their previous semester(s) at Columbia

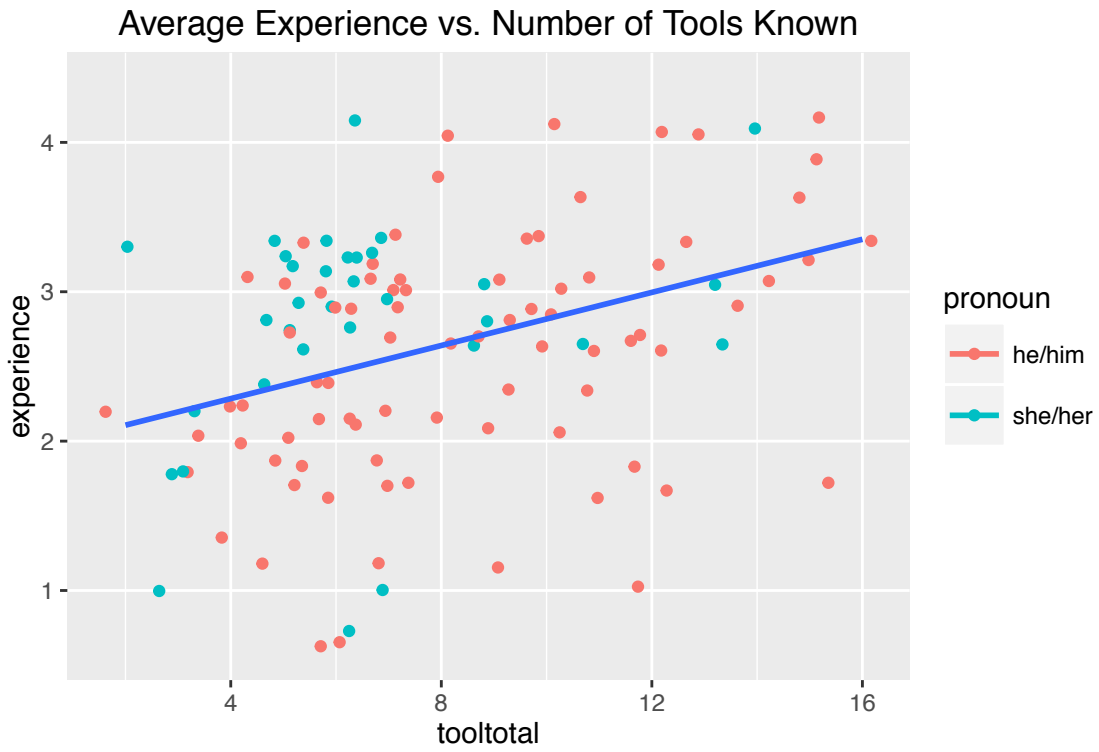
The lack of color purple in all graphs demonstrates that level of 4 is almost or completely non-existent, which means almost no one deems him or herself as an expert in a particular tool. This is probably ideal because students who were experts would probably not be able to significant improve their skills in this class.

C. Correlations

The next few plots try to analyze whether there are correlations between various different variables. Some plots show a positive correlation, whereas other indicate there is not relationship between the two variables.

1. Total Tools Known vs. Overall Experience

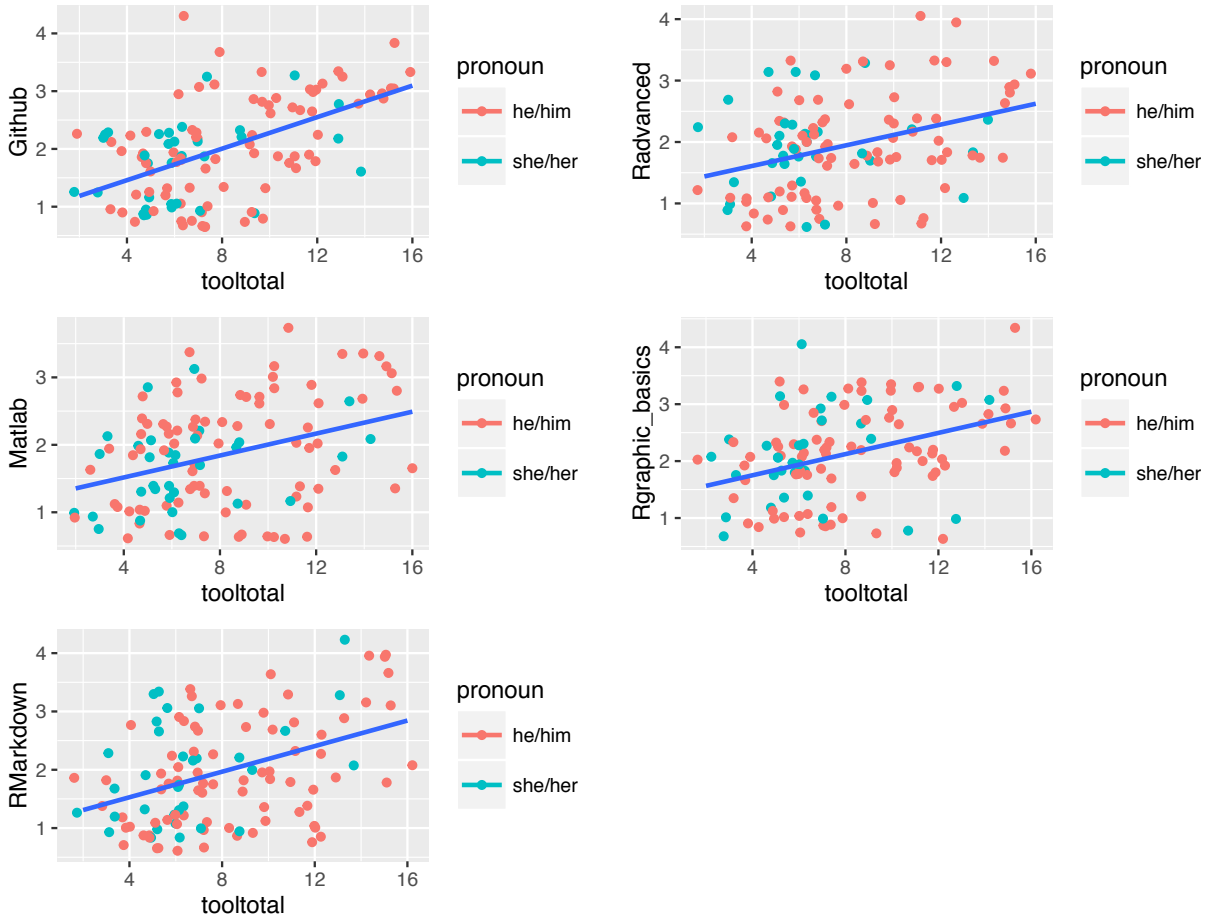
Our next graph plots students overall average confidence against the number of tools they know. Unlike the graph above, there does seem to be a relationship between these two variables, with an increase in the number of tools a student knows corresponding to an increase in their overall average experience.



2. Experience Level in Individual Tools vs. Total Number of Tools

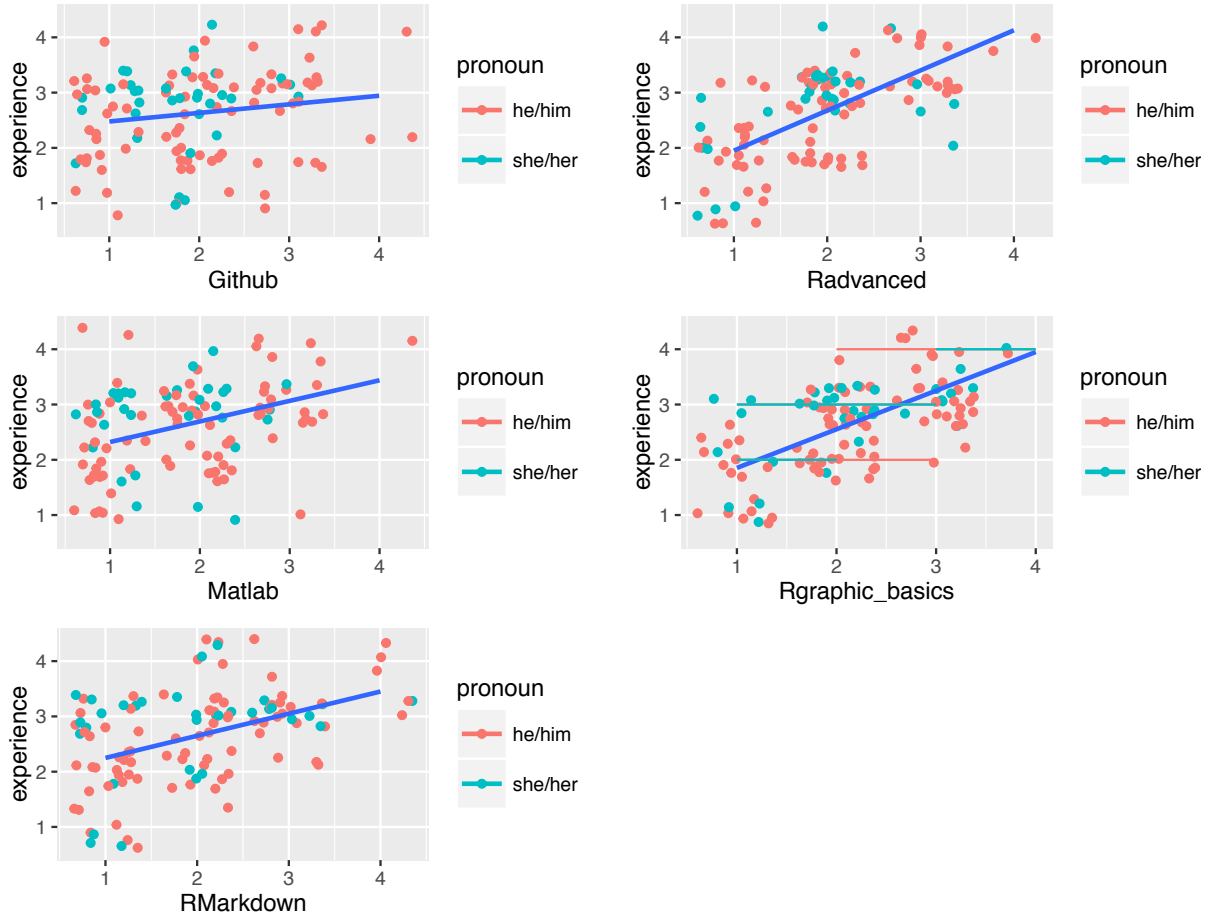
This multiplot set compares each student's experience level on the specific tool with the number of tools that student is able to use proficiently.

It is easy to detect that all linear relationships have positive slope. Thus, there is a positive relationship between the number of tools a student knows and their expertise in the specific tools examined.



3. Experience Level in Individual Tools vs. Overall Average Experience

The next plots show the relationship between a student's experience level in a specific tool and their overall experience level.



We can see that the relationship between experience with Radvanced and overall experience is comparatively larger than majority of other relationships shown in our multiplot set. In other words, only a small increase in the overall level of confidence in R-advanced tools will lead to a comparatively large increase in the overall level of confidence.

4. Pairwise Comparison Between Six Tools

Our last figure is a multi-pairwise comparison plots among all six tools. Since we have 5 tools (Github, MatLab, R-Markdown, R advanced, and R graphic basics, we exclude the "R data manipulation and modeling" due to the fact this is not a specific tool but an overall evaluation of R skills), thus the total number of different plots will be

$$4 + 3 + 2 + 1 = 10\ plots$$

Since all levels of experience are numerically transformed from the original categorical dataset, the transformed columns of tools' experience levels are also categorical. Within the ggpairs() package, there is no option for jittering points; therefore, we manually add lightly-influencing

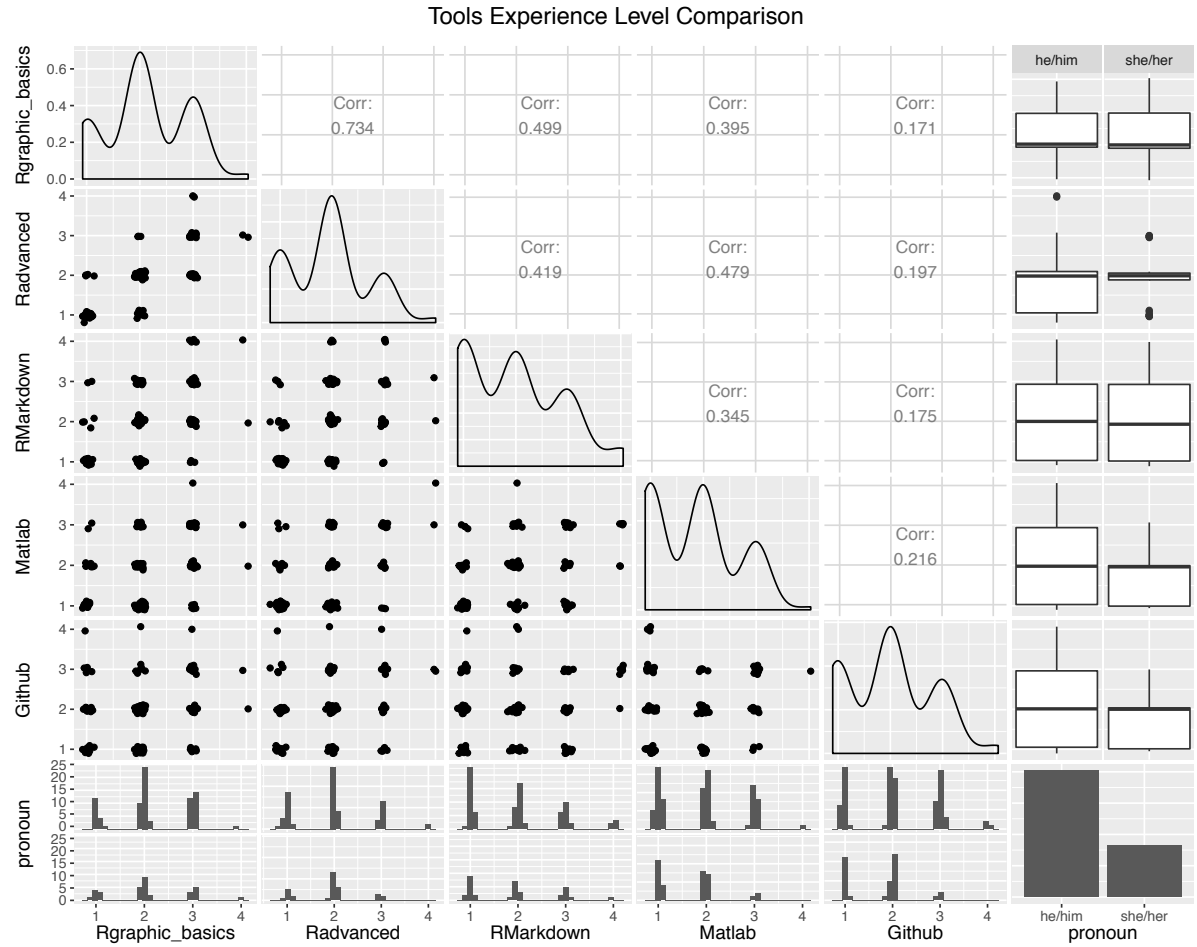
noise into our new data-frame of all 5 tools and the corresponding levels of confidence. Due to the fact that when adding noise ε with

$$E[\varepsilon] = 0, \qquad and \qquad Var(\varepsilon) = 0.05$$

the noise will spread around 0 with variability of $\sqrt{0.05}$ unit, the overall relationship between the original level values will not be influenced too much as the noise will balance out among itself. The original correlations are:

$$Cor(R_{grap}, R_{adv}) = 0.731, \qquad Cor(R_{grap}, R_{mkd}) = 0.508$$
$$Cor(R_{grap}, Matlab) = 0.402, \qquad Cor(R_{grap}, Github) = 0.173$$
$$Cor(R_{adv}, R_{mkd}) = 0.416, \qquad Cor(R_{adv}, Matlab) = 0.482$$
$$Cor(R_{adv}, Github) = 0.203, \qquad Cor(R_{mkd}, Matlab) = 0.338$$
$$Cor(R_{mkd}, Github) = 0.179, \qquad Cor(Matlab, Github) = 0.219$$

which are really close to the ones after manually jittering our original data points. One of the main reasons we add this noise is that we want to reduce overlap between our data points, which will give us more obvious visualization of the relationships among these tools' confidence levels.



From the upper triangle of our plot-matrix, there are 10 values of correlation coefficients between each two tools. Except for the value between R Graphic Basics and R Advanced, it is easy to detect that majority of the coefficients are less than 50% which implies that among most tools, learning one of the tools may not be **helpful or necessary** for learning the others,

especially for R-based software and Github/Matlab. However, since R Advanced contains numerous high-level technique tools in R software, proficiency in R Advanced therefore benefits learning all the other R-based/related tools, such as R Graphic Basics.

III. Conclusion

To conclude, we wanted present possible future analyses that we believe would be interesting extensions of our work with this dataset.

First, we wondered how we could place students in groups so that the teams are as diverse (in terms of skills) as possible. If we use a learning algorithm, then it would have to be unsupervised because there are no labels in the data that could be used to train the model to split the class into such groups. K-means clustering and hierarchical clustering come to mind, and have the advantage of us being able to choose the number of teams (total number of students / 7 students per team). Unfortunately, both of these methods would form teams based on how similar the students are. For example, one group may have students who are all experts in RStudio, another team may have only Python users, etc. To deal with this problem, we can compute a single score for each student, say the total number of tools that they listed, as an overall measure of skill level. We want to partition the class into groups of size 7 such that the average skill level for each group is maximized. This is an optimization problem, and one way that it can be solved is by applying the dynamic programming algorithm from the following paper: http://arxiv.org/pdf/math/0309285.pdf .

Second, since this survey was carried out at the beginning of the class, it would be helpful to send out a similar exit survey before the end of the course. By comparing the two surveys, we can determine whether students learned more tools, are more confident in the tools that they have already used, and whether they prefer to use any other text editors (for example, migrating from Eclipse to Sublime). We can break these results down in terms of gender and academic program to answer additional questions: Have female students become more confident in the tools and techniques they already knew, such as RMarkdown or visualization in R? Have students from certain programs been able to learn tools with which they might not have otherwise had experience, such as Applied Math or M.S. in Statistics students learning more about GitHub or SQL? This would be a great way to assess what students learned from the course, and whether being placed in diverse groups helped their learning.