Investigation of EDAV Skills and Programs

Team - No Free Lunch

February 11, 2016

Project Description

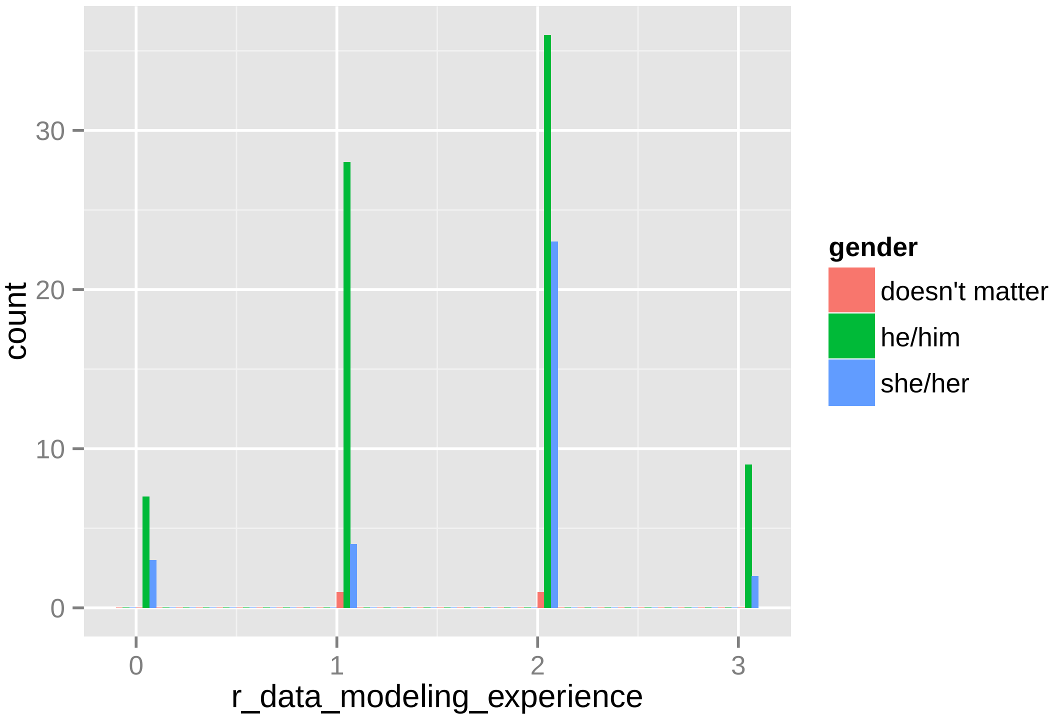
Within this document we will be showing multiple visualizations and explanations of skills that the Spring EDAV 2016 class have. We begin with intial plots to gain an understanding of what the population of the class looks like and then work towards more complex views of similar subjects. In the end we will work to attempt at predicting the program of a test group of students by looking at their skill attributes.

Basic Info

Here we see the mean of students' reported skills levels. There were 114 total students, and levels ranged on a scale from 0 (experience "none") to 3 (experience "expert"). We can see that students reported being most familiar with R data modeling, and least familiar with matlab:

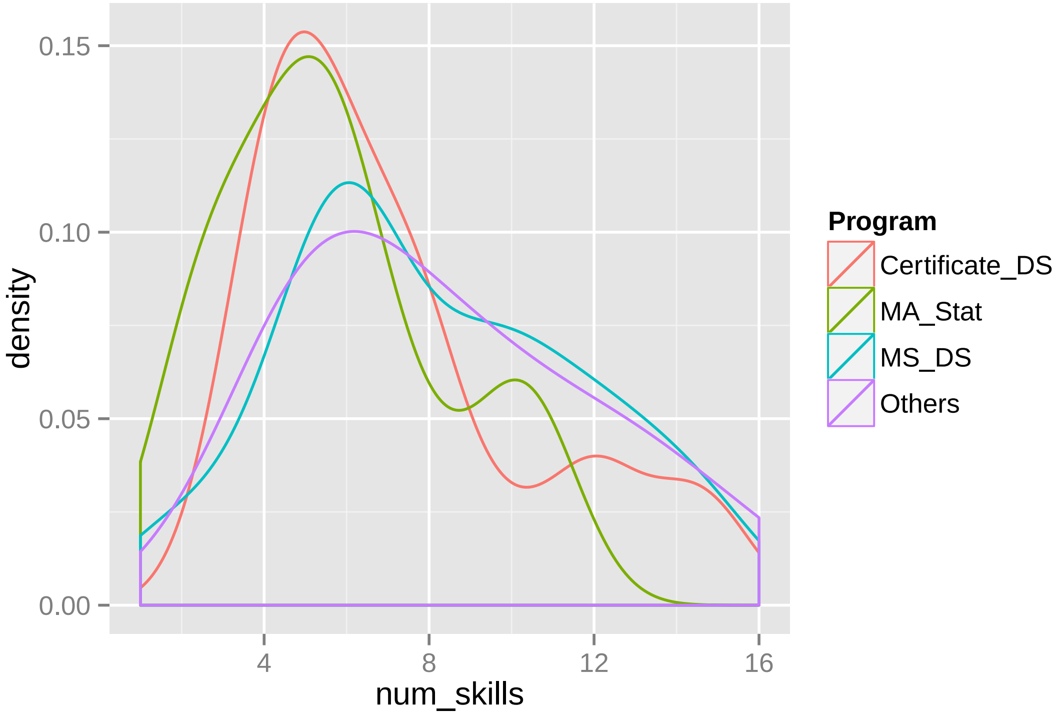
Matlab	GitHub	R Markdown	R Multivariate Analysis	R Graphics	R Data Modeling
0.833	0.991	0.956	0.939	1.114	1.632

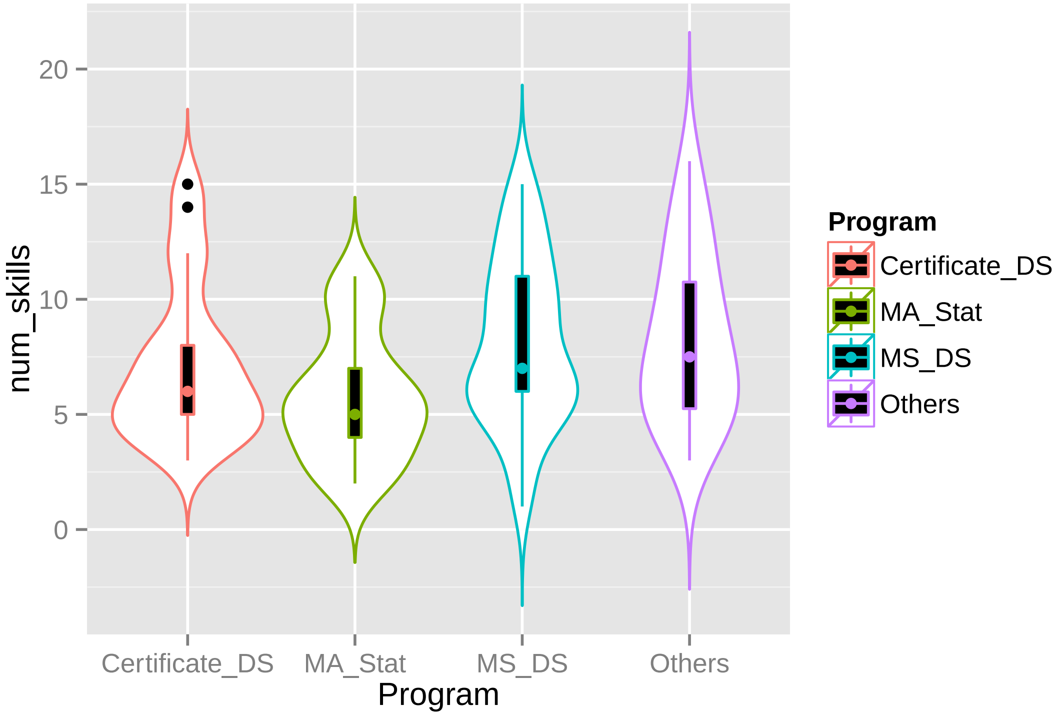
R Data Modeling Experience by gender Here we see a breakdown of students' R data modeling experience by gender. We see most of the students have reported experience levels of 2, and are "confident" in their R data modeling skills.



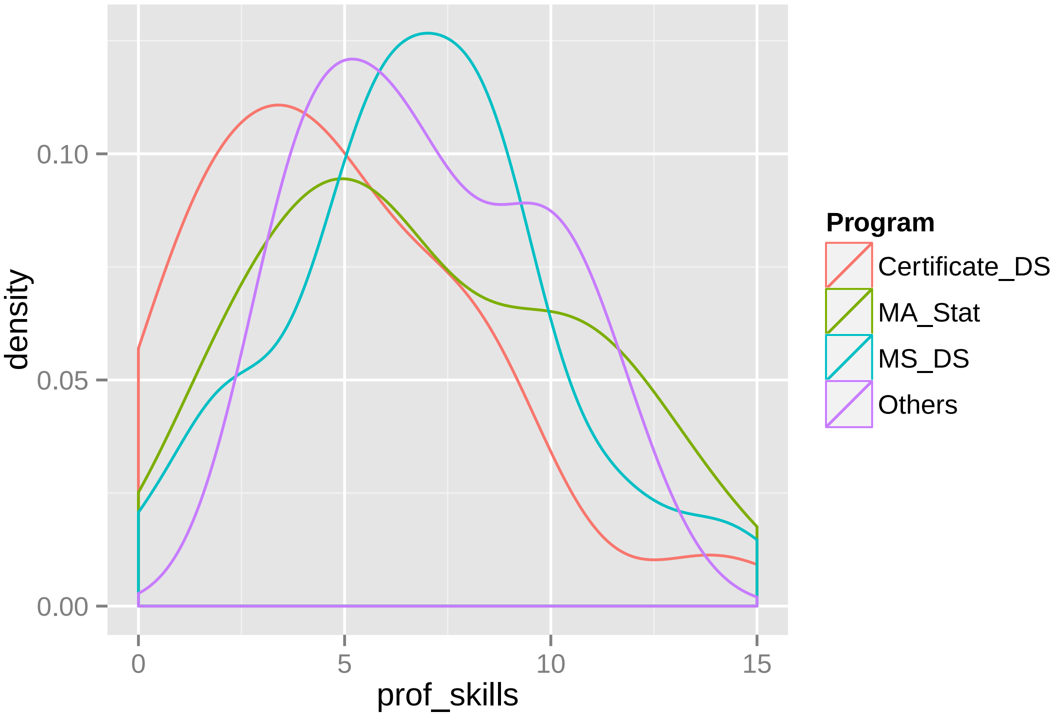
Further graphical analysis

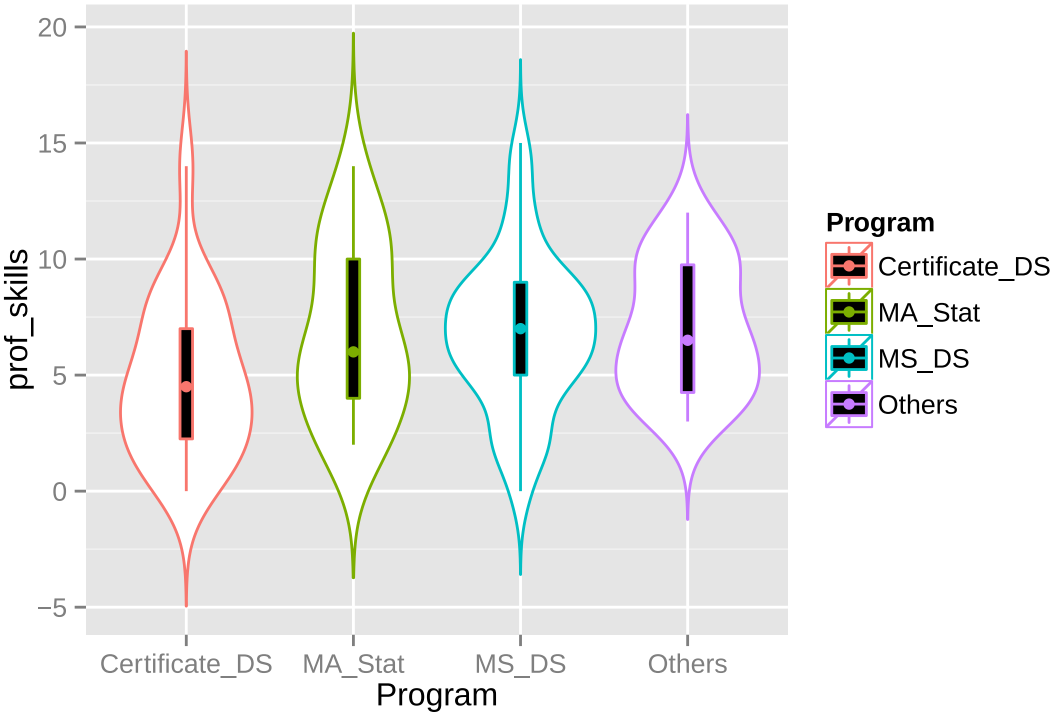
Use ggplot to draw kernel distributions, boxplots, joint distribution by contour of num-skills and prof-skills for different programs. This will help gives us a better understanding of the breakdown of our class.



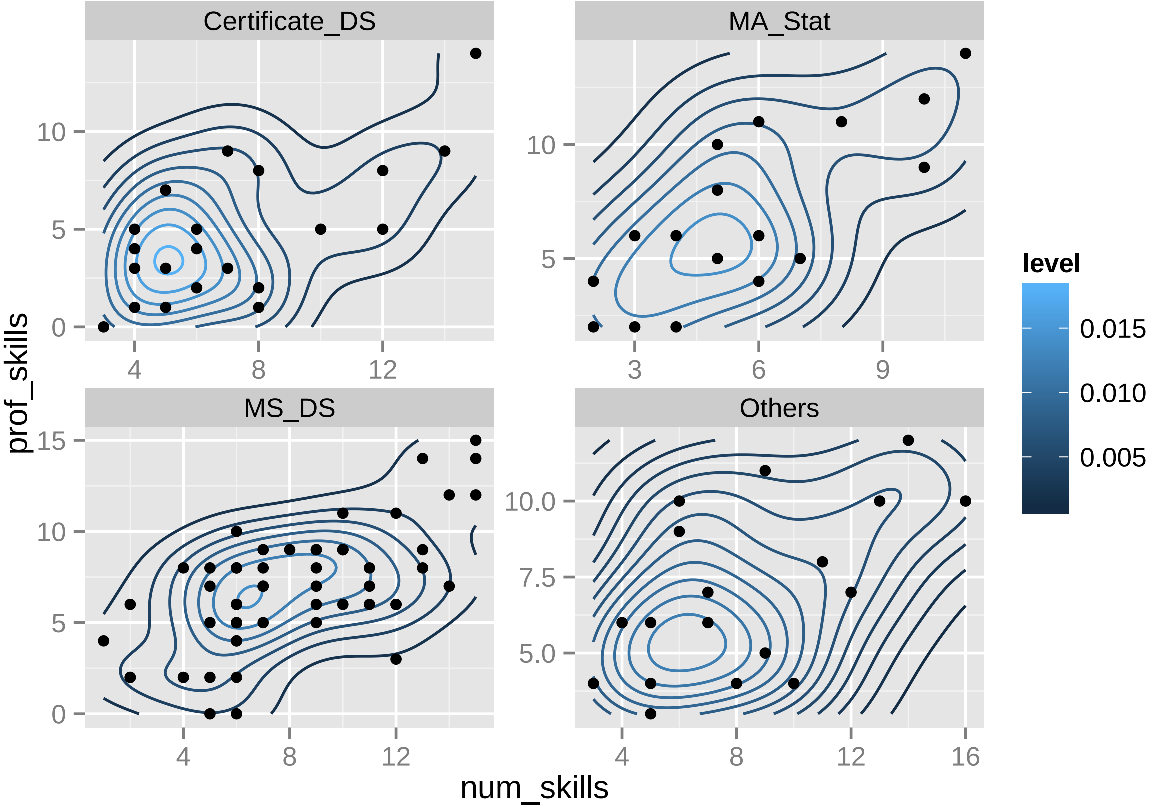


These two plots reflect the distributions of the number of skills of students from different programs. We can see that these four distributions are all skewed to the right while the distributions of Data Science Masters(MS_DS) and students from other programs(Others) have long tails.



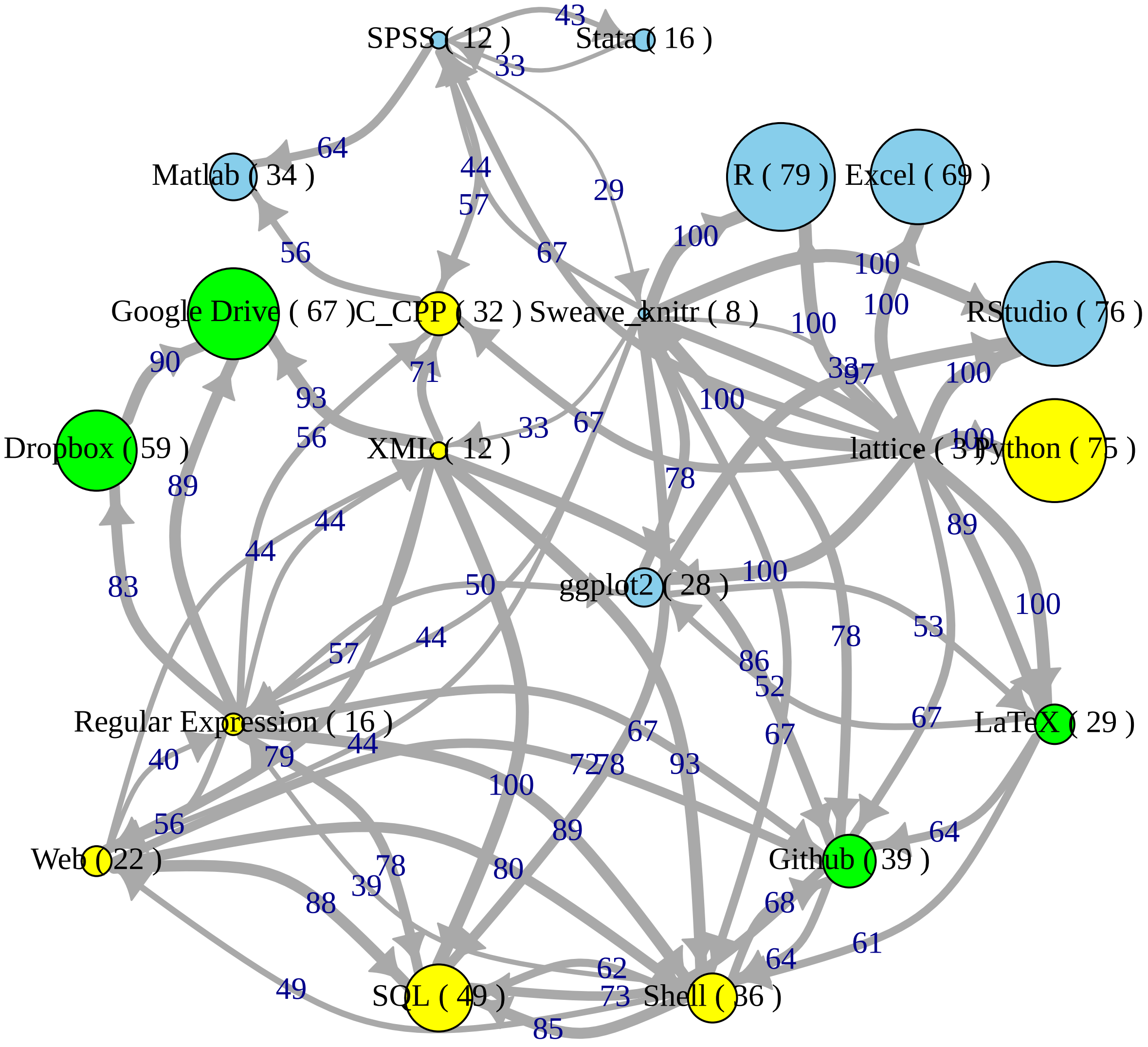


These two plots reflect the distributions of the proficiency of skills of students from different programs. Interestingly, Data Science Certificate students tend to report less proficiency of skills than other three groups in terms of median of the distribution. The distribution of students from other programs has short tail. Besides the distribution of Data Science Masters(MS_DS) tends to be pretty normal distributed.



This plot reflects the joint distributions of the proficiency of skills and the number of skills in different student groups. Contour lines represent density of this distribution. From this joint perspective we can see that the distributions of Data Science Certificate students and Data Science Masters(MS_DS) have larger density around “peak” than other two groups.

Network graph on percentage of having each skill



This graph shows the information about percentage of students who chose each skill and the relationship among skills.

Each vertex represents a skill. The vertex label shows the name of the skill and the percentage of students who chose each skill among all students. The size of a vertex represents the percentage of students who chose it. So, the higher the percentage is, the bigger the vertex is. The color of vertex represents the area of the skill; we classified each skill into three areas: Computer Science (Yellow), Statistics (Blue), and General (Green).

Edges represent the positive relationships among skills. For example, the probability of a student in this class knowing SQL is 49% (represented by the number on SQL vertex), but among the students who reported Web as a skill, the probability of also knowing SQL is 88% (represented by the label of the Web->SQL edge). This can be written as Pr(SQL|Web) - Pr(SQL) = 88% - 49% = 39%. We can say that students who know Web skills have a higher probability of also knowing SQL than the overall probability of knowing SQL in this class.

So, each edge A->B shows the probability Pr(B|A) as labels in percentage unit. The thickness of edge width graphically shows how high the Pr(B|A) is. The thicker the edge is, the higher the Pr(B|A) is. Note that the relationships among skills are not symmetric, because Pr(B|A) - Pr(B) is not necessarily the same as Pr(A|B) - Pr(A).

We only show the strong and positive relationships where Pr(B|A) - Pr(B) > 20%. For example, the percentage of students who chose RStudio is 76%, while the percentage of students who chose RStudio among whom chose R is 90%. The difference between these two probabilities (Pr(RStudio|R) - Pr(RStudio)) is 14%. So, even though 90% seems very high, because the difference is smaller than the threshold (20%), there is no edge from R to RStudio in this graph. Even though it wasn't shown on the graph, it is interesting to know that only 90% of students who chose R also chose RStudio. We can guess that they probably have used R console or other text editor.

By looking at edges focusing on vertex colors, we can see that skills in the same skill area (e.g., Computer Science) tend to have more edges among them than between different area skills. General skills such as Google Drive and Dropbox seem to have stronger relationships with Computer Science skills (Yellow vertices) than Statistics skills (Blue vertices). This may be because students with computer science background tend to use these skills more often than students with other background.

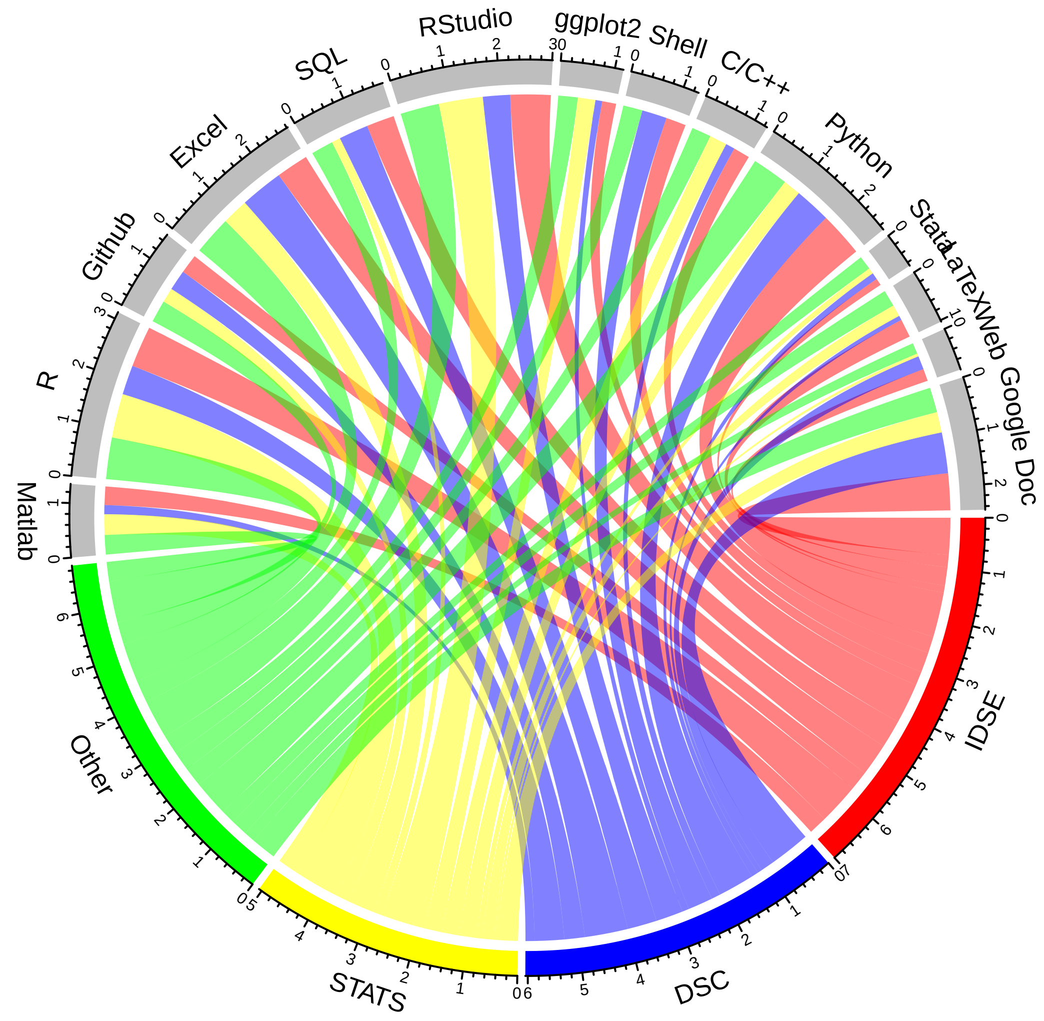
Another interesting characteristic is that rare skills (represented by smaller vertex sizes) such as Sweave_knitr, lattice, and Regular expression have relatively thick edges with many other skills. A possible explanation is that these skills are rare because they are more advanced skills, and it makes sense for students who have advanced skills to have many other skills as well.

In the case of Python, it has strong relationship only with lattice. This might be because the majority of students are data science masters and data science certificate students, and the majority of them took an algorithm class last semester where they used Python. So we think Python did not show any strong relationships with other skills, because the majority of data science students used Python regardless of their previous experience or background.

Fundamental graphic analysis on skillset

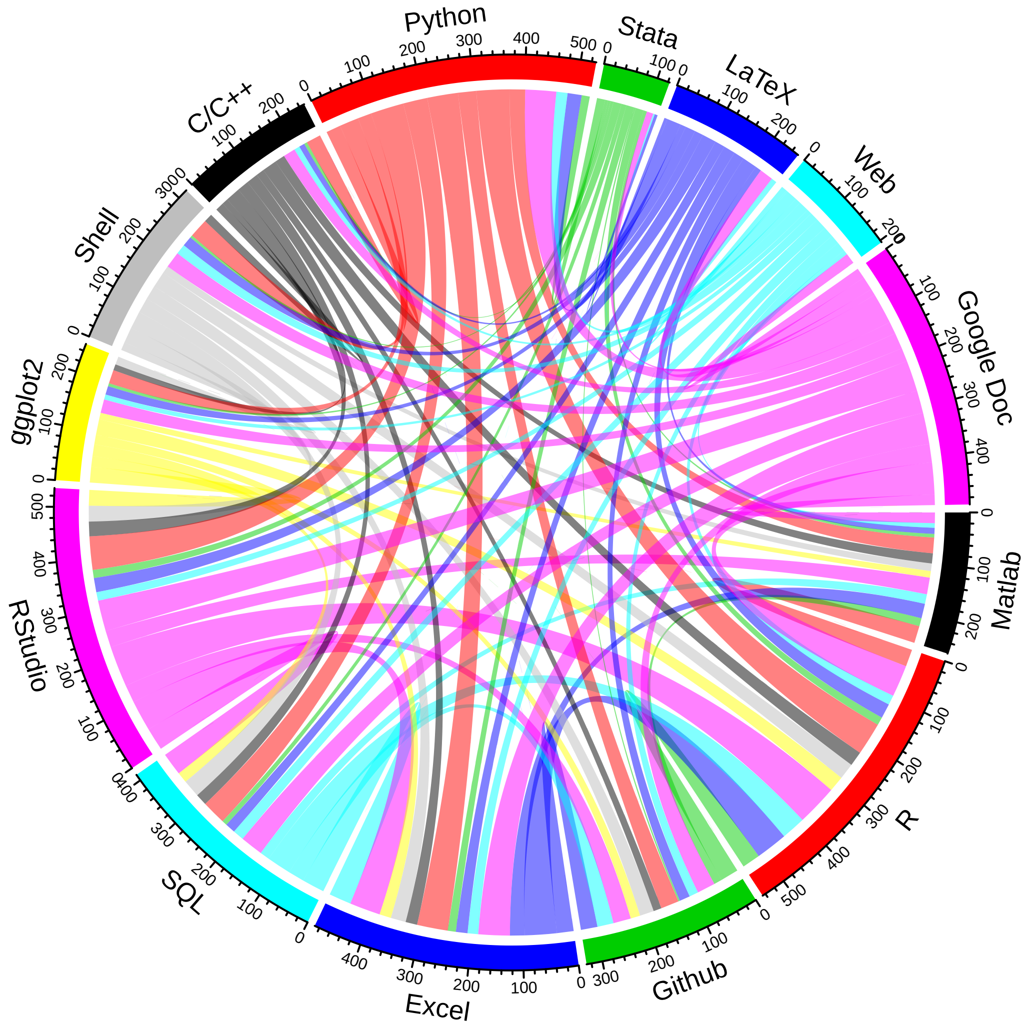
A chord diagram could illustrate intuitively the relationship of skills, i.e., the proportion of people who have a skill (e.g. SQL) also have anther skill (e.g. Python). Also it would be good for visualizing the relationship between skills and program of people. Thus it would provid us with basic guidance towards deeper analysis.

Visualization of program-skills relationship To visualize this relationship, we need to selection features (columns corresponding to skillset questions in our case and program column), split each skill into one new column as bitmap (e.g. if 1 in SQL means familiarity for SQL and 0 means not). So the cleaning scripts as describe in previous sectors are used. Here df_clean is further extracted and transformed into our desired data frame.



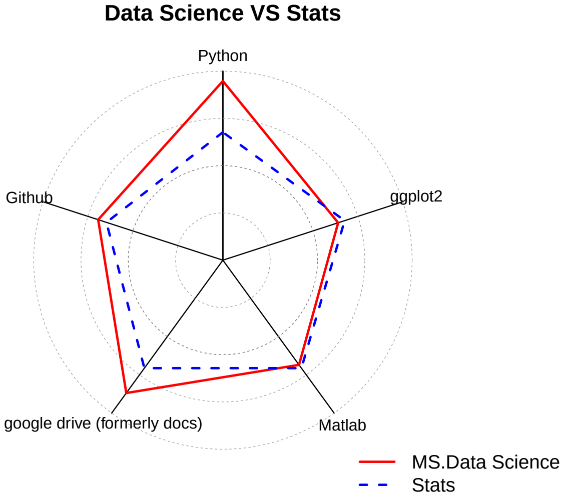
So the chord diagram for program to skills is created. Each degree has a corresponding arc in the circle and each chord (the colorful thick lines inside the circle) connects a proportion of students in each program to their corresponding each skill.

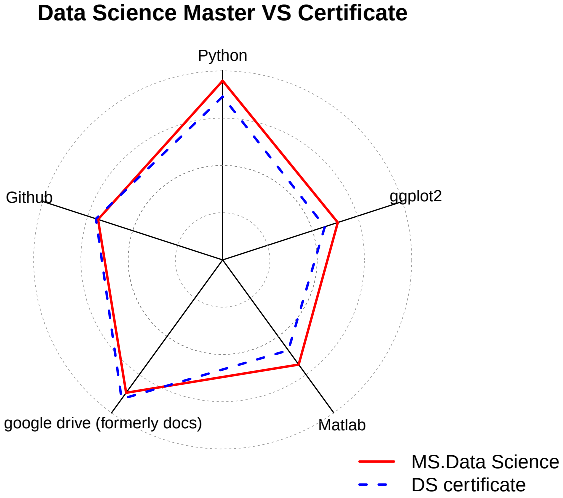
Visualization of skill-skill relationship Then we further transformed the dataset for creating a new chord diagram showing the relationship skill-skill relationship.



Differences between Programs - Experience with Tools

We created a visualization that shows the information of experience with tools across majors. The visualization is called radar plot. It will help us to compare majors through the difference in experience with tools. The graph will show for each tool what is the proportion of people who know how to use the tool.





radar plot ref : http://www.statisticstoproveanything.com/2013/11/spider-web-plots-in-r.html

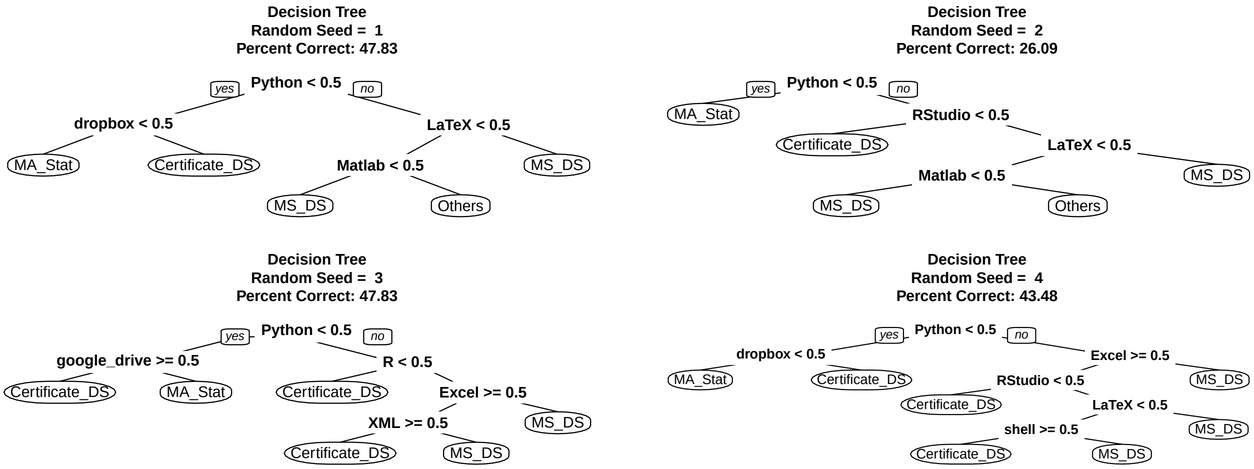
There are two plots made. One compares Stats with Data Science and the other compares the Data Science master with Data Science certificates. We chose these these majors because the majority of people are in these majors. We can compare as many majors as we want if needed. We chose the top five tools identified in the random forest classifier we made for classifying majors. On the plot the further the line is from the center, the larger the proportion of people who know how to use the tool. The range for variables is from 0 to 1 (proportion). We can clearly see that the patterns for Stats and Data Science are different. The pattern for Data Science master is similar to the pattern for Data Science certificates. These results show that Stats students are different from Data Science student on experience with tools. While Data Science students' experience with tools is similar to that of Data Science certificate students. Also, we can use this kind of plot to see what are people from different majors good at.

Decision Trees

We will now look at a decision tree to try and understand if we have the ability to predict what program a student is in only the student's experience with the software programs and tools listed in the survey.

A decision tree was chosen because the intrepetability is high and can give us some insight into what categories help create the purest subgroups using the Gini Index

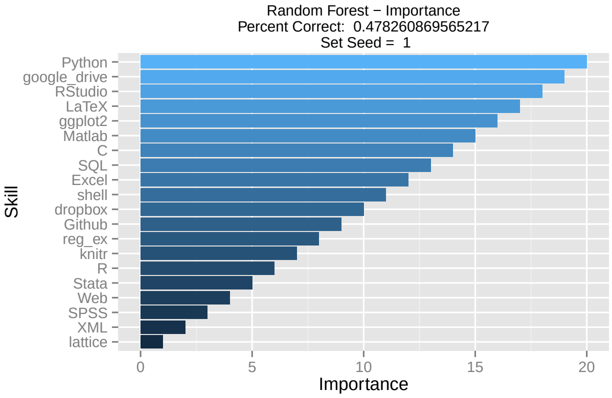
The training set is set to 80% of the given data and attempt to predict on the remaining 20% to get an idea of how this prediction algorithm might perform. We will also change the randomness of the selection of the training set by using `set.seed()`. This will allow us to see how high the variance might be for the tree. If the tree greatly changes based on different training sets we are experiencing a high variance.

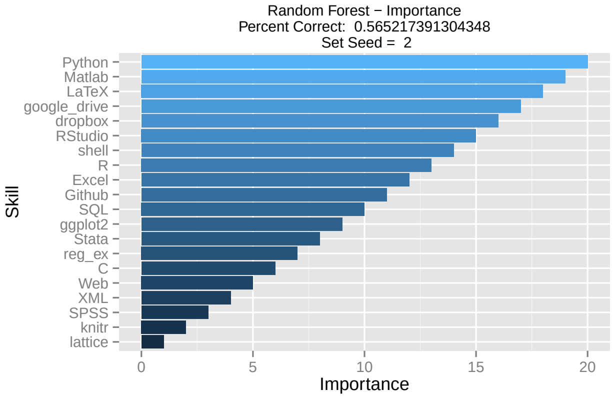


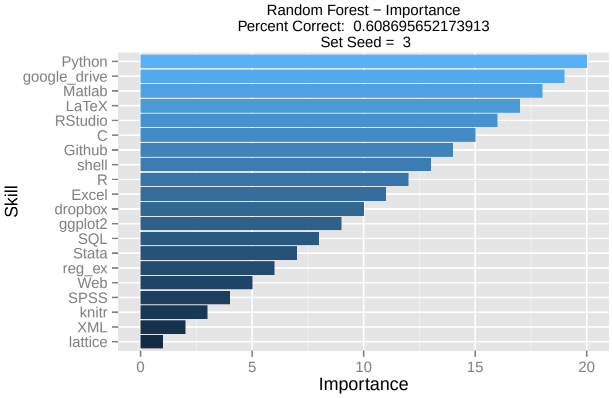
We can see from the training data the tree has selected "dropbox" for the first split in all four cases. While hte trees are not necessarily performing well we can see that the trees have changed after the first split in every case. This signifies a model experiencing high variance. One way we can work to bring the variance down is using Random Forests, which will randomly select the first split over many decision trees and use a voting process to determine classification. This voting process will decrease the variance that we are currently seeing and should improve overall performance. We will see a benefit as long as the decrease in variance is greater than the increase in bias that will be experienced.

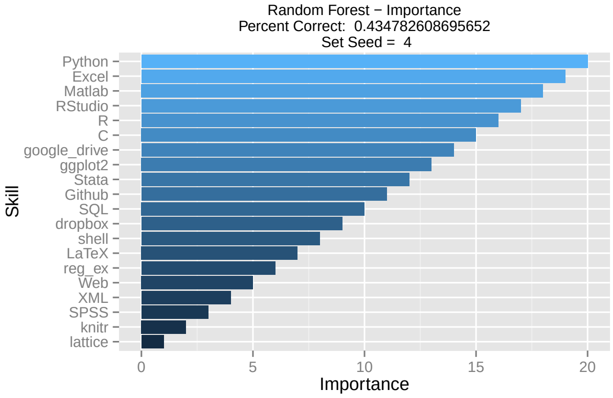
Because of the small dataset Random Forests trains very quickly so we can run multiple training attempts very quickly. We will look at a range of trees to use in the random forest and see how it performs. Random Forests have the a trade off of higher accuracy but harder interpretation than typical Decision Trees.

For the following we will look at accuracy and Importance. Where the imporatnace calculated using the mean decrease in the Gini Index over all of the trees. In more simple terms these graphs show the most important variable that causes the purist division within the data.


Random Forest − Importance
Percent Correct: 0.478260869565217
Set Seed = 1

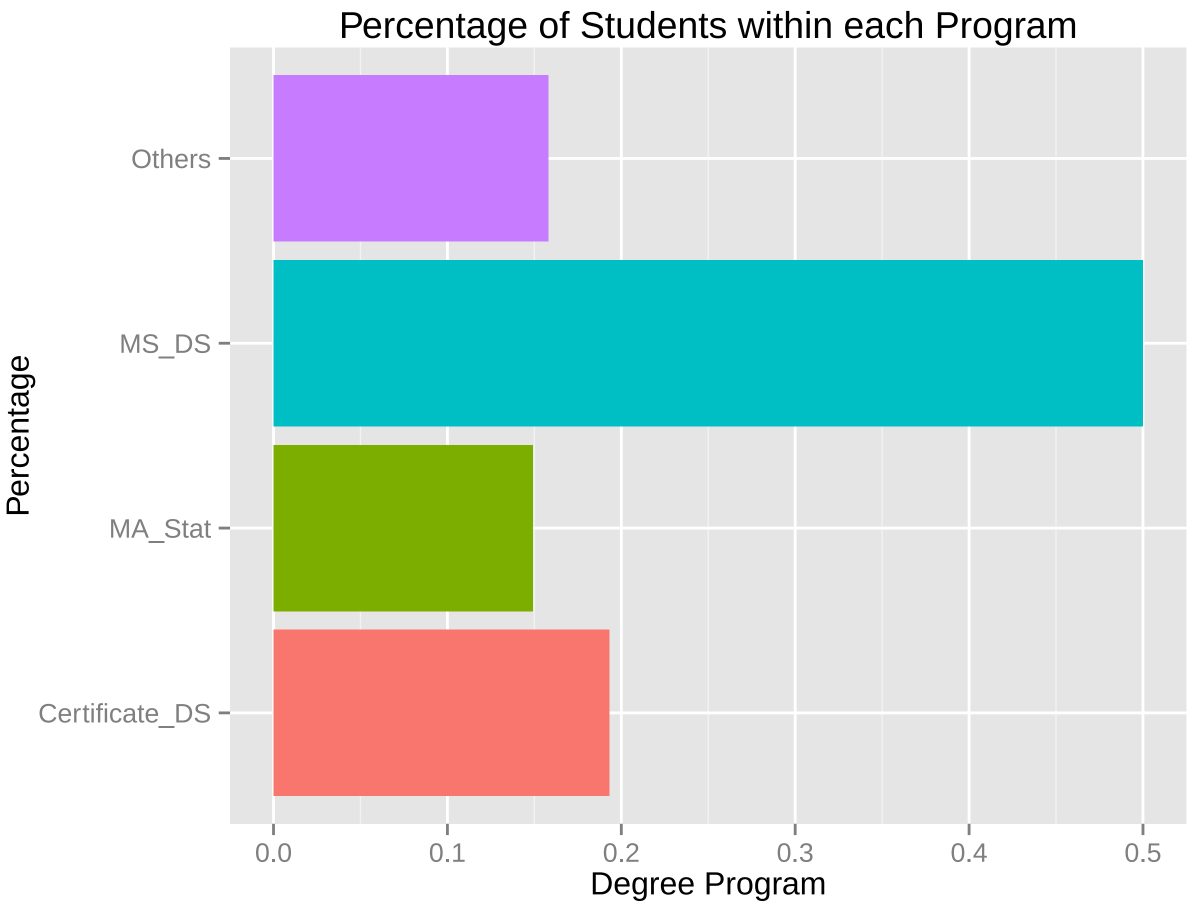

Random Forest − Importance
Percent Correct: 0.565217391304348
Set Seed = 2


Random Forest − Importance
Percent Correct: 0.608695652173913
Set Seed = 3


Random Forest − Importance
Percent Correct: 0.434782608695652
Set Seed = 4

Overall we can see a increase in the performance of the predictions with different trees. While we saw a decrease in when `set.seed() = 1` for all of the other case we saw an increase as expected. Most likely if we were to recieve more training data we could expect to improve on our prediction. On average over this small training set we were 52.17% accurate.

To understand how we did we can look at the percentages of all of the majors



Overall the largest major within the class is IDSE (master) at 50%. So if we just consistently guessed IDSE we could still do fairly well. The Random Forest only did slightly better at 52.17% on average.

Looking into decision trees helped us gain a better understanding of what factors might help differentiate the programs. We saw consistently that dropbox seemed to play the largest factor in how our algorithm determined which student belonged to which program. However even with 200 trees in the Random forest we still saw the importance ranking change, suggesting our data is still very spread and has a high variance. This is where more data could help the performance.

Conclusion

Overall within this document we explored many relationships between Program and Skillset. We were able to take multiple views of the data and begin to understand the complex relationships that occur between skill and Program. This helped us gain a better understanding of the distribution of the class and how the different divisions of programs correlate with different skills.