

Results

Out of the 42 responses from the Survey, 31 people are comfortable with "statistics" and 30 people are comfortable with "probability." This makes sense because this is a statistics course and most who are taking this course have taken Stat 133, 134, and/or 135. That being said, it is interesting to see how many people are comfortable using R with a measly total of 2 people. This tells me that although we should assume that statistics majors make up the majority of an upper-div statistics class, not all have taken Stat 133 or 135 which involve the use of R in their curriculum.

A third of the class is comfortable with LaTeX, which is useful for when we need to make our presentations look nice and around one-third of the class is comfortable with blogging, Github, and XML. It is nice to know that some are comfortable with Github so that they can guide the other two-thirds of the class along as we will use Github throughout the entire semester. Again a confusing point arises. If people have taken Stat 135, then they have (hopefully) used XML on their Google Earth Projects. This data conflicts with only two people having used R before this class. All of this, however, is to assume that the data we collected is complete and correct (basically that people treated this survey seriously). It may be possible that not everyone was clear on whether you could check one box or multiple boxes within the survey and this could lead to a lack of data. I don't think it is possible that only ONE person in the entire class took stat 134/135/153/154 but the survey shows that only one person checked that box.

Since there are 9 people who know how to use iPython notebook and Python, I say that these people can be the new technical leads for newly assigned groups along with the people who know how to use github (there may be some overlap). This will be until the rest of the class fully learns Github and knows how to navigate through iPython notebook.

As far as the learning style scores are concerned, there is a broad range of each kind of style with no dominant style that the class majority favors. However, there seems to be more people who tend to learn better kinesthetically than any of the other three learning styles. With my past experience, I know that learning math requires practice and programming is the same. Learning by doing works better than listening to a lecture about how to code because once people try to apply what they hear in class, problems arise.

Roadblocks

Some roadblocks that we faced weren't severe and we didn't have any that we couldn't overcome. Dave could not use his virtual machine on his computer because of a RAM issue. Since he could not participate on the programming side of the project, he was the one who took charge and emailed the group to meet together. He also provided a list of roadblocks below.

Other roadblocks involved viewing the work that each of our group members did. For example when Khoa uploaded his work on his ipython notebook, Christina and I were confused on how to access his work from our laptops. We also had many errors in creating graphs in R and transferring them over to iPython and through Github.

Along with this we had problems accessing the Google spreadsheet. The latter problem was solved by creating new bConnected Keys opposed to using old ones that connected to Airbears2. The graph problem was solved by connecting the group through IRC and talking out our problems with sending data back and forth through Github. For example Christina had some errors on her virtual machine that she didn't know how to fix. She typed out the problem on IRC and Khoa helped resolve the problem.

Dave's part:

- 1) Communication: To effectively communicate in code we created our own channel on IRC (#statpacker)
- 2) Technical: My Mac did not have enough gigabytes to host a virtual machine and run IRC, Chrome, terminal, Adium, and related applications to the course. So with the help of my group mate, I installed Ipython and its related dependancies on my Mac until I find a long-term solution, for the data will only increase in size.
- 3) Technical: One of our members (visualizer) had a difficult time downloading the repository from Github, so another one of members(technical) sent her the code so that she could generate the graphs
- 4) Communication: The presenter was not wholly aware of Aaron's directions for the assignment. The curator, therefore, sent him an e-mail, delineating that the process is equally as important as the end result. Aaron emphasized that he would prefer imperfect presentations with reproducibility, than a perfect presentation with no transparency.
- 5) Communication: Our technical is well-read in computer languages. He helped our visualizer install matplotlib and its related dependencies so she could visualize the cleaned data.

Reproducibility

Our technical lead Khoa created an iPython notebook page with step-by-step instructions on how to reproduce the results that we ended up with. The process involves giving an instruction, showing the code, and then showing the results of the code. By following the guidelines shown in the notebook, one could easily go through each step and produce the same or similar results that we produced.