Transcript for 11/16 Webinar

Note the transcript has been only partially checked for accuracy so please see recording: https://youtu.be/XwCccbvMBvQ

Teaching Reproducible Research

Inspiring New Researchers to Do More Robust and Reliable Science

Karl Broman

Mine Çetinkaya-Rundel

Benjamin Baumer

A FREE webinar featuring:
Karl Broman, University of Wisconsin (@kwbroman)
Mine Çetinkaya-Rundel, Duke University (@minebocek)
Moderator: **Benjamin Baumer, Smith College** (@baumerben)

Wednesday, November 16, 2016, 2:30–3:30 p.m. EST
Twitter Hashtag: #ASAwebinar

ASA Sponsors: **ASA-MAA Joint Committee, Statistical Education Section, Statistical Learning and Data Science Section**

With recent emphasis on robust and reliable science, a minimal standard for data analysis and other scientific computations is that they be reproducible—that the code and data are assembled in a way that all the results can be re-created (e.g., the figures in a paper). While adopting a workflow that will make results reproducible will ultimately make a researcher's life easier, this goal will not be easy to achieve without the right tools and organization.

In this webinar, three reproducible research experts share how they teach undergraduate and graduate students to make their research reproducible. They recommend instilling best practices in students as early as possible and teaching data analysis at all levels of a science curriculum using a completely reproducible framework. In this way, new researchers will know no other workflow than a reproducible one. They also urge statisticians to marshal efforts to promote reproducible data analysis practices in other disciplines. While all this might sound like a tall order at first, modern tools for literate programming (e.g., R Markdown) and systems for version control (e.g., GitHub, Open Science Framework) paired with carefully designed curricula make this goal easier to attain than ever before.

Note the transcript has been only partially checked for accuracy so please see recording: http://magazine.amstat.org/videos/education_webinars/ReproducibleResearch.mp4

Speaker 1: Welcome everyone. The webcast is about to begin. Please note today's call is being recorded. Please stand by.

Ben: Thanks everybody and welcome to this ASA webinar on reproducible research. My name is Ben Baumer. I'm an assistant professor of statistical and data sciences at Smith College. We are very lucky to have Karl Broman here and Mine Cetinkaya-Rundel as well, and I will introduce them more formally in a second.

This session is about teaching reproducible research, Inspiring New Researchers to do More Robust and Reliable Science. My first order of business is to thank some of our co-sponsors. These include the Center for Open Science. Center for Open Science among other things is one of the developers of the Open Science frame work which is an online management system for doing reproducible research. This session is also co-sponsored by the Peer Review Evaluation Group. They help to facilitate greater transparency in the process of peer review which is very important to all of us in doing our reproducible research.

I'm going to talk for a couple minutes, and then I want to give Karl and Mine plenty of time to talk to you about reproducible research. We're going to come back at the end and do some Q&A. If you have questions, please feel free to write them into the question box that you see on the webinar and we'll answer those as they come in or at the end of the session.

One of the reasons this came about was my involvement and Mine's involvement in an organization Project TIER. This is Teaching Integrity in Empirical Research. This is a group run by Richard Ball who is an economist at Haverford College. What they've done is put together a protocol for how to do a reproducible research project. Their goal is to try get more and more students in the social sciences and the sciences onboard with that protocol and doing reproducible research. You can see their coordinates here. For those of you who are faculty members out there, I would encourage you if you're interested in incorporating reproducible research into part of your work to think about Project TIER and maybe doing a faculty fellowship with them.

This webinar in particular came from an invited session of the joint statistical meetings that happened this past summer which Mine and I co-organized on this very same topic, obviously. This was a well attended and popular session. If you missed it, Karl has compiled a collection of slides from actually all across the joint statistical meetings, so not just this session but all the sessions and in particular this session. You can see there's a link there to his GitHub repo that has some slides from that talk.

I just want to talk a little bit about this idea of statistics as a vehicle for making science better. I think we have seen some progress in the last few months even in terms of greater transparency in terms of the kind of data journalism that has become so

popular. In particular, Five thirty-eight has put out a GitHub repo that now contains lot so data sets and even some codes for some of the articles that they've done. It's not everything, but it's certainly a step in the right direction as far as this issue of transparency and reproducible research.

Lastly, how could we not mention last week's election. In my class, we talked about some of the predictions that were made an din particular on the slides here you can see predictions that were collected by the upshot of the New York Times. It turns out in retrospect that one of the things that differentiates some of these models, the 538 model in particular which has most pessimistic outcome for the Democrats, was the correlated errors across states. It appears as though that's exactly what we saw in the election, that the polls were off, but they weren't off independently. They were off in such a way that they all moved in the same direction.

I can only imagine that our ability to understand this process would be enhanced if we knew more about how all of these models worked. Some of these models are more or less public, others less so, and I think our speakers will address these issues and some others related to this notion of doing reproducible research.

I want to introduce our speakers, and I'm going to introduce Karl Broman who's going to be our first speaker. He is a professor in the Department of Bio-statistics and Medical Informatics at the University of Wisconsin-Madison. His research is in statistical genetics, and he is the developer of the R/qtl package. Karl got his PhD. in statistics from UC Berkeley and was a faculty member in the Department of Bio-statistics at Johns Hopkins for many years before joining the University of Wisconsin-Madison. Karl was also named a fellow of the ASA for his contributions. Karl is going to talk for about 20 minutes. Our second speaker is going to be Mine Cetinkaya-Rundel, and I will give her a more thorough introduction at that time. Karl, please take it away.

Karl: Thanks, Ben. I'm very excited to participate today. I hope you all feel free to ask questions in the chat area. If you think of something later, feel free to contact me by email or Twitter.

I'm an applied statistician working largely on genetic problems. I have a lot of collaborators and really enjoy helping people make sense of their data. I spend a lot of time looking at data and writing reports to my collaborators describing what I've done and what I've learned.

Not too long ago, these analyses were often a bit of a mess, at least behind the scenes. After many painful experiences, I've put a lot of effort into revising my approach and into my work life to have things be more organized and reproducible. By reproducible I mean that the code and data for a project, for an analysis project, are assembled in a way that you could hand it to someone else and they could re-run the code and get the same results, the same figures and tables.

I'll start with an example. This is an email I got from a collaborator in response to an analysis report that I had sent to him. He writes, "This is very interesting. However, you

used an old version of the data and N equals 143 rather than N equals 226. I'm really sorry you did all that work on the incomplete data set." You get an email like this and my initial reaction was “Why am I using the wrong data file and where is the right data file?” What he didn't know is that I had adopted a reproducible workflow so that I spent 20 minutes trying to figure out where the right data file was, and then 10 minutes getting it incorporated in where it was supposed to be and type one command and re-run the all the analyses and reproduce the report, and because things didn't change substantially I could with a half hour's work send him right back a revised analysis with the full data. This is a reproducibility success story, maybe a small one, but the key thing is that I could recover from this mistake and without much work fix it.

The second feature here to note, I think, is that I tend to when I write an analysis report I start with a paragraph that describes the data and what I view the goals of the analysis to be. By doing that, he could see right at the top of the analysis that N equals 143 he's using the wrong data set. If I hadn't included that brief summary of the data at the top he would never have known that the whole thing, and I would never have known that I was using the wrong data set.

This is what I strive for. My life still isn't always so rosy. Often, I'll get an email from a collaborator that the results in table one don't seem to correspond to those in figure two, or I'll come back to a project after three months and I'll look in the set of scripts, the R scripts, that I've written, and I'll say what order am I supposed to run these scripts? Or a key data file for analysis project nobody really knows where it came from. Like I had one project where the key annotation file for a gene expression microarray is just something we got from some person in the past and we don't really have any kind of documentation of where it came from.

Or in some late night exploratory data analysis effort, I will decide that I need to omit some set of samples because they are badly behaved, and then six months later when we go to write the paper I'll realize I need to explain why we omitted those samples. Can I now do a bit of forensic analysis on my analysis to figure out how are those three samples different so that I can explain why I chose to drop those and not something else?

Or in some late night bouts of exploratory data analysis I'll make some cool figure that really looks like a super interesting gene and my collaborator's going to love, and then the next morning I realize I didn't write down what gene it was or how was it that I made that figure. The worst thing is maybe coming back to a project and learning that your script is now giving an error, can you trace back and figure out I know it was working three months ago why is it not working now.

I sent an email to someone about a paper where I was really interested in trying to use their method and compare it to my own. The response I got back was “The attached is similar to the code that we used” which is not a position you really ever want to be in.

Reproducibility is assembling the data and code for a project in a way that you can hand it to someone else and they can re-run the analysis and get the same figures and tables,

the same results back. We differentiate that from replicability where you gather all new data and do analysis on those new data and come to the same conclusion. Some scientists have used these terms exactly opposite, what statisticians have come to call reproducibility we often think of as replicability and vice versa, but it's important to distinguish those things, and we're really seeking a minimal standard for computational work, that with exactly the same data and code, they're assembled in way that someone else can re-run then and get the same answers. This is also different from the results being correct. The analysis could be fully computationally reproducible, but there maybe is a bug in the code, or in your understanding, so that the results are totally wrong. So, reproducibility is assembling the data and code in a way that others can re-run it and get the same answers. Sort of a minimal standard. We strive for more, but at least we want to get that.

In getting from the standard practice to a fully reproducible workflow, lifestyle, is a difficult task. Looking back at my history of trying to improve my approach to analysis, I wrote a webpage talking about what I call "The steps towards reproducible research," and that's what I want to walk you through today. You can read it in somewhat more detail later.

The first step, I would say, is to organize your data and code. Jenny Bryan, a professor at University of British Colombia, she has a great quote, "File organization and naming are powerful weapons against chaos." Another quote I really like, paraphrasing Mark Holder, "Your closest collaborator is you six months ago, but you don't reply to emails." So, the first you thing you want to do with a project is to organize the code and the data so that if you come back to it three months later, that you can look in there and really know what everything means. Everything's in the right place. It all makes sense. I think in doing just this one step, you will have made your analysis project reproducible in the way that you can hand it someone else and it will be understandable.

The basic approach I take is that every analysis project is in a separate directory, and the directory for one analysis project is split up into sub-directories that are meaningful, and they tend to be exactly the same sort of directories for every project I work on. For example, I separate the data from the code. I have a folder that is just the raw data, and then other folders that have code that I'm using. I also really prefer to separate the raw data that I get from my collaborator, from any derived data that might come from the project. I'll have maybe, a separate directory that has notes to myself, or references that my collaborator has given me that are related to the project. I try to have also a "read me" file, that really describes what the project is about. And where I can, in each of these sub-directories, I would have a "read me" file that explains what the separate files are. If you've gotten a project into this form, you hand it someone, and it will all be clear, so that they can at least, in principle, read what you've written, look at the files, and be able to re-run the analysis.

Differentiate that from what I call "chaos," as Jenny Bryan said. This is a project folder on my hard-drive, for one project, I'd say that this is the project that led me to reflect on my approach to things. That ... Yeah, you don't want a project directory to look like this, where you have say, you know, folders that are Ping, Ping2, Ping3, Ping4, and you know,

Int2_for_Mark, and such things. It can be hard to organize a project, and it's also hard to keep a project organized from day to day. I like that think that, I guess, from experience it seems that, the organization of a data analysis project is dependent on really the worse day that you've spent on it. Everything's kept well organized and looking great, and then one day the collaborator says, "I have a grant due next week. I need you to do this, and that, and the other thing." And you go in there, and all kinds of craziness occurs. Then it's all going to be a big mess from there on in. So, organization is really the first thing you want to strive for, and it's something you pretty much have to work on every day. And, there you are. But that is a big step towards reproducibility.

The second major step I recommend is, everything that you do with data, you do it via a script. You know, some computer program. That, if you need to open up an excel file and save it as CSV, you should do that really with a script. If you get an excel file that has 18 worksheets, and one of them has the column names a little bit differently, you're tempted to maybe go in and fix that one column name to match the other 17 worksheets. But my mantra is, if you do something once by hand, you're going to do it a thousand times. So really, everything you do with the data, you want to be through code. A lot of things that are really kind of difficult with code, but you want to strive towards improving your scripting ability so that you can do everything through code.

If you've gotten these first two steps, everything's organized and clear and documented, and secondly that everything is with a script, then you have really a fully reproducible project, that you can hand to someone else and everything you've done is all there through a script, they can, in principle, run everything.

The third step is to try to automate the full analysis process. I like to use an old tool, GNU Make. Make is a command line tool that was originally written for the compilation of large software projects. You know, you have a bunch of different C or Fortran files that need to be complied to object code and then link them together. Make was really originally written for that purpose. But, you can use it to automate really any command line driven analysis project. For instance, here … the way it works is you have this one text file that describes all the things that you want to create, and then what other files they depend on. So, the final product of this analysis is a webpage, and it depends on these files, the original R Markdown file, and some clean version of the data is a comma delimited file. So you have a target, and then what files it depends on, and then you have a line of code that says what you need to do to turn the dependent files into the target to create that final analysis html file.

The advantage of make is that it both automates the whole process of an analysis. So, turning raw data from an excel file to CSV file, and then doing some initial R scripts to do some cleaning of the data, and then producing an analysis report; it automates that process. Then it also documents the dependencies, documents the set of things that need to get done, what it's produced from, and how it's produced. Make can be a bit quirky. At the beginning of each of the lines that says what command gets run, there has to be a single tab character, not a set of spaces. And if you're going to change directories within that command, you have to do it sort-of all on one line. If you change directory and then in the line below if you were to run R, it would jump back into the original

directory. Make is a bit quirky. It's an old program, it's complicated. But, for automating and really documenting the process, I still find it really useful.

If you've done those three things, you have everything organized, you have everything in a script, you have a make file that governs how everything gets done, then that's really a fully reproducible analysis. You can hand it to someone else, they can type make, and redo everything.

Step four I would say, is rather have everything through scripts, you want to focus on creating reproducible reports. I use, and I like R Markdown for this purpose. This is an example report that I wrote a couple years ago. At the top of the report I have some summaries, like there's 36,813 markers, and there were 1400 or so phenotyped mice, and 1500 genotyped mice, and so forth. All those numbers are coming straight from the data, rather than me typing them in. Behind the scenes, there is this R Markdown document that has bits of text with bits of R code inserted, and when this document gets processed, those numbers get inserted in place of the code used to generate them. That is really what has made my life much happier.

The next step that I would recommend, are to really look at the code and try to make it better and more readable. The first thing I would do to try to make your code more readable, is to turn any kind of repeated code into functions. You know, in Python you would use this def statement to write a function, which you might write for reading in a step of data, where in R, you might use the function “function” to create a function. But, if you look at an analysis report, you often have a bunch of chunks of R code making different plots. In many cases you'll have, you know, you've written a bunch of code to make one plot, and then you want to make a similar plot, but with, you know, plotting from different columns or something. Often you will copy a chunk of code down and then do some editing, and you'll do that repeatedly, several times. Repeated code makes things harder to maintain. If you decide you want to change the overall look of the plots in your analysis, or there's some other change to those bits of repeated code, you have to go and change all those chunks of code. If instead, you write a function, it makes it so that if there's something you want to change, you just have to change it in that one place.

Secondly, it makes your analysis report easier to read. Instead of having these long chunks of code that are doing the plotting that you want, you have replaced that with a single function call, you know, “plot genotypes”. Or, you know, each chunk of repeated code is replaced with a well named function call. One way to make your analysis scripts or R Markdown documents, your reproducible documents more readable, is to take chunks of repeated code and turn them into functions, and replace those chunks with calls to those functions.

The sixth step would be to take those functions that you've used, and turn them into a package or module. R packages, you think of R packages like ggplot, or dplyr, I mean, these big projects written by people that really know what they're doing. In fact, R packages are not too terrible to create, and even if you're making a package for your own use, you're not intending to distribute it to anyone else, having code that you use in

multiple projects sit together within a package in some common place on your computer, can really make a lot of things easier. But, rather than, if you think about a plot that you've made in the past that you want to make again, you figure out, "What project was I working on? What was the time of year that I was working on that that I made that plot? That one little nice plot?" Rather than have to sift around for it, if you have in advance made a package of code that you like to use, for yourself, you can find it in that one place, and it's easier to reuse code between different projects.

The seventh step towards a happier life is to adopt a version control system. Everyone that uses a computer has to, at some point, deal with different versions of files. We all have some way of keeping track of different versions of files. Often, it's this way, following this comic, phdcomics.com. You write a paper, you sent it to your professor, and you revise it, and you go back and forth, changing the name of the file to reflect the new version. This can be effective. If there's one thing maybe to learn from this it's that, you should never use “final” in a file name. You're sure to revise it later. Here's another directory on my hard drive, a whole bunch of files. You'll see a bunch of them that have final in the name, and you'll also see final-old, right here, or final-revision2, which is my favorite. Don't use “final” in a file name. We need ways of keeping track of different versions of files, and it's hard going at the beginning, but adopting a formal version control system like Git, and its web home for many people, GitHub, can really make your life easier in the long run.

The way Git works is, basically, you're keeping track of one project directory and all of its sub-directories. When you change files for a project, you commit to those changes and record a little message of what it is you've changed. Having done that, you can then look back through the history of changes in a project and be able to see, you know, on this date I changed this file by deleting these lines and adding these additional four. Version control really shows its advantage when you're working collaboratively on a project, in that if you and a collaborator are both working on a common set of files at the same time, merging your simultaneous changes are really easy with a version control system. It also allows you to go back to the state of a project at any point in the past. If a script is no longer working, and you know it was working three months ago, you can go back to the state of your project three months ago and verify that it was working, and then you can step forward and really see what it was, where was it that it stopped working, and what had I done at the time it stopped working. Adopting a formal version control system is an investment of considerable effort, that will pay off in the long term.

People often think of version control strictly for software projects, and especially big software projects, and that's a great use for them, but pretty much everything I do on the computer, at this point, talks I write, papers I write, analyses and software, all those things, my webpages, they're all within Git, and using GitHub.

Finally, the final step is … reproducibility is about assembling a data and code for a project in a way that you can hand it someone else and they can re-run the analysis and get the same results, the same figures and tables. If you're going to hand the code to someone else, you need to explicitly license the software. The way copyright works in the US, you automatically own the right to distribute and perform the code you write,

and it's only by explicitly telling people via a software license that they're allowed to do the following things with this code, that they can. If you want to be able to hand your code to someone else and have them run it, you need to pick a license. Pretty much any license. Pick a license. Software licenses usually do two things. One, they say what people can do with the code. That they're allowed to run it, and modify it, and redistribute it. Secondly, a software license generally will protect you in case something terrible occurs. It will usually say, "If something terrible occurs, don't blame me." So, you want to license your software really for those two purposes, that telling people what they can do with the code, and so that if it breaks something, they shouldn't sue you.

Finally, I want to end with a quote from Keith Baggerly. It's talking about reproducible research. He said, "The most important tool is the mindset, when starting, that the end product will be reproducible." That is really the key. You sit down at a project, and if at the start you say, "I want this to be reproducible," that's really the key for ensuring that you will stick to the effort of making sure that it is reproducible.

That's that from me. My slides are online. Feel free to continue to ask questions in the chat area, or if you want to contact me later, find me on Twitter, or you can find my email on my webpage. Thank you.

Ben: All right, thank you Karl! So, I think we'll discuss questions at the end. There's a couple of good questions in here already. I guess I would pitch this one at you. A member of the audience, "So, the potential problem with using GitHub is that you can not copy data with personal identifiable information there. For me, in industry, the solutions have to be local or on the corporate cloud at most." In your experience with GitHub, Karl, do you have any advice about people who have personally identifiable information?

Karl: I would say, you can use Git locally. So, even if you want to keep your code just to yourself, the code and data to yourself, it is worthwhile using Git for version control within your own computer. I think, when you're putting code and data on GitHub, you do need to be really careful about it, and make sure that the data itself is kept locally, even if you're posting the code on GitHub. In industry, I would say that many look for other options. There's a GitLab software that a company can use in house that has many of the features of GitHub, but is kept entirely within the company, if there's data that they don't want to share more broadly.

Ben: So, that's a good point. Git and GitHub are not the same thing. And right, that Git is an open source program that you can use locally without the GitHub web interface.

I want to welcome our next speaker and we'll have time for more questions for Karl at the end. Our next speaker is Mine Cetinkaya-Rundel, she is the Director of Undergraduate Studies and an Associate Professor of the practice in The Department of Statistical Sciences, at Duke University. Her work focuses on innovation in statistics pedagogue, with an emphasis on student centered learning, computation reproducible resource, and open source education. Mine is the most recent winner of the Waller Education Award, for her contributions and innovations to the teaching of elementary statistics. So, I'll leave it to Mine, and we'll be back in a few minutes to take some more

questions.

Mine: Thank you very much, Ben. Today I'm going to be talking about some of these ideas that Karl talked about in terms of reproducibility, and how to introduce them, actually weave them, through the undergraduate statistics curriculum. I feel like in terms of getting people to do better science, and reproducible data analysis is part of that, we need to take a two pronged approach. The first prong would be to convince current researchers to adopt a reproducible research workflow. That is, I think, a harder feat, because that's talking to people who already have an established work flow. What I tend to work on more is what I think is a slightly easier feat, which is training new researchers who don't have any other work flow. So this is thinking about our students who are teaching data analysis for the first time, and how can we do that such that they learn the best practices, as opposed to the sloppy practices, if you will.

The idea of reproducibility often comes up in the context of published research, and the need to accompany this type of research with the complete data and analyses, including software and code. We hear about reproducibility a lot as an issue in terms of journal submissions, for example. But as a statistics educator, so I teach data analysis, I believe that it is our responsibility to instill these best practices in the students before they set out to do research, so that when that time comes for them, they don't really have any other work flow, and they are bound to the reproducible work flow that they have learnt. So, what I'm going to talk about today is how we try to accomplish this within the statistics curriculum at Duke. I'll give examples from three of our courses, and then talk a bit about what more do we want to do, or might you want to do, the tool kit, and also some of the additional pleasant side effects that come from teaching within this work flow.

The first course I'm going to talk about is a traditional intro stats course. Maybe it's not your most traditional ones, but in terms of the traditional topics covered, I would say that's your stat101 type of course. It's the first course for non-majors. It's not calculus based. It's mostly social science majors taking this course. And it is possibly the only quantitative science course these students take. So I feel like, we need to do a very good job teaching these students, both in terms of statistical reasoning, and also data analysis, the best practices here. In this course, how does reproducibility come into play? On a weekly basis, students work on computational labs, and also the work on a data analysis project, at the end. So, doing hands on data analysis is an integral part of the course. The way students do this in this course is through literate programming. They are doing their analysis in R, and we introduce R in this course not so much as a programming language, but more as a statistical data analysis tool. They interface with R through our studio, mostly because of how easy it makes it to create these reproducible reports using the R Markdown package.

To give you an example of the work flow that I'm talking about, in a traditional setting, in such a course, data analysis might be completed in a graphical user interface based program. Such as Mini Tab, or SPSS, or something like that. In that case, the data analysis, so the descriptive statistics, plots and tables, the model output is generated in this software, the statistical analysis software, and then write up is done elsewhere. In a

text editor, like either Google Docs, or Word, or whatever. So that's where the research question and context comes into play, the calculations and conclusions. Then, to generate the final lab report, the students need to combine these two through a copy past paradigm, into a lab report.

And this is error prone, first of all, because it is possible that the analysis gets updated, but the wording does not, or vice versa. It's also open to kind of mucking the results, if you will. Because, there's not necessarily any guarantee that the copied and pasted plot is actually generated by that student, or has not been altered since it's state that came out from the data analysis tool. A better paradigm that we work with in this course is where both writing, and the data analysis, so that is the writing of the code, as well as the output, all happen in the same environment. So you're test blocks and data analysis all happen in the same environment, that's the R Markdown file, and every time this document is rendered, the code is re-run again, and the text is re-rendered again. So, you are guaranteed that the final report actually has the most recent version of everything that you're looking for. So, for example, assume that the data comes in a CSV file, so that's one component of the students work, and the work happens in this R Markdown file, and when rendered, it generates a html file. It could also be a word or pdf file, but we have the students generate an html file, and submit that simply through the course management system. So that's Sakai for us at Duke, but it could be Moodle, Blackboard, whatever your institution is using.

That's where the feedback is done. So, the faculty member or the teaching assistants can provide the feedback there. Just to give you an idea, here is what the students have submitted. It's their R Markdown file, as well as the generated html output. Simply using the interface of the course management system, we provide them some feedback. The nice thing here is that, we could simply look at the resulting html file and only view the results, but especially if there is a need to reproduce the students work, to either check that is is, indeed, reproducible, or if there is a mistake and we simply want to change something and fix it to see what would happen if that one small mistake was fixed, the fact that the students also submit the R Markdown file makes this incredibly easy to do so.

Often times the question that's asked is, "Can students handle this?" There's already, a lot going on in a stat 101 type course, can they handle this additional complexity? I'm going to say, yes, that they can. The most important being that point-and-click is no less overhead than scripting. So, I happened to pick one point-and-click touch software, Fathom. This is a software, it's pretty neat actually, it will allow you to generate graphs by simply dragging and dropping variables onto the axis and tables, similarly. Suppose that you wanted to make a contingency table with row or column proportions. This is verbatim text from a lab handout where students are shown how to do this, how to generate this contingency table with row and column proportions. This is no less work for a student to parse through, than two line of R code where the syntax might be new to them, but that's where the teaching happens. We teach them the syntax. So, I would say that, I don't think students necessarily prefer the point-and-click software, if it still takes this much explanation to get something done.

Additionally, in this paradigm, we have the code and output always together. That's very useful for instruction, but it's also incredibly useful for students. It removes one burden off of them in terms of keeping things organized. Syntax highlighting that happens by default in an R Markdown file, makes it easier to learn the language a bit because, especially for a visual learner, it's nice that functions are colored a certain way, and arguments are colored a certain way. Finally, it keeps code organized and work space clean. So, errors due to overwriting a data file, for example, happen much less often, if ever, when you're working in this paradigm. Those are the types of errors that are very difficult for novice users to recognize, so it really does help actually remove the frustration from the student.

Another example I'm going to give is from an Introductory Data Science course. Here the difference, it's a first semester undergraduate course, it's again, not calculus based, but the students here are interested in quantitative sciences. In fact, this is likely the first of many of the quantitative science courses these students are going to take. They're probably going to major in something like stats, CS, or math. In this course, in addition to what I described earlier, we also use version control tools like Git and GitHub. So, what's happening in terms of the work flow here is that, the generation of the data analysis is done exactly the same way, through literate programming, but students can actually then instead of using the course management system, put their work on GitHub, and they can use the GitHub environment to collaborative with each other for team projects, for example. When they're finally done, the assessment is also done on GitHub. So an instructor or TA then, can pull their work, provide feedback there, and push it back for them to get the files back and do any sort of enhancements, or just review the feedback.

Here is what a submission looks like. In this case, we have one repository per assignment, either per person or team, depending on what type of assignment it is. These are private repositories. GitHub is pretty nice in terms of providing private repositories for educational use, so we don't have to worry about both students seeing each others' work before it's time to do so, and also any additional FERPA considerations. You can see that it's easy to see, in this team, how many times people have contributed. If we were to click on this, we could drill down and see which contributions are made by which students. And they are, instead of emailing files back and forth to each other, they're simply committing their work to the repository, and pulling from there when the next person is ready to work on it.

Again, the question, can students handle it? And yes, they can. But, I would say that introducing this additional wrinkle of using Git and GitHub, it does come at a cost in terms of instruction time. I would say that if you are doing this, it is important that you make time in your course structure to allow for instruction of how this work flow works. It is, I think, unrealistic to assume that students, especially those who are new to programming, that they will themselves figure out exactly how everything works, in terms of using GitHub. And one nice thing, in terms of the tool kit, is the R studio interface with GitHub makes it a lot easier for the students. They don't necessarily need to know any command line. And they can get, I would say, 98% of the way there, in terms of what they have to do. It is possible that they will get themselves in a space

where you might have to step in and save them, from any GitHub error that they might be getting, but I think that a majority of the time, the fact that it works, and it provides a very clean space for them to collaborate, and also submit their work and get feedback, we would get them used this very useful tool that they can take with them and use throughout their undergraduate career, as well as beyond.

The last course I will mention is a course that comes later in the curriculum. This is a second or third year elective. It's a statistic computing course. It has some statistical pre-requisites. This course is for students who are committed to the stats major or the miner, and it is either the first or second computing course these students take. Also, what they learn from this course, we anticipate that they will use it in their future work, as well. In addition to everything that we described here, in terms of reproducibility, we also use additional build tools like make, that Karl mentioned earlier in the talk. The reason for introducing this additional tool is that, now the projects that the students are working on are not necessarily ideal candidates to be housed in a single markdown document. The complexity is higher, statistically speaking, computationally speaking, and even potentially in terms of the tools that they're using. They might be using languages beyond R, for example, to do part of their work. So, something that allows them to then regenerate all of the work, that isn't just tied to an R Markdown file, is useful for some of the complex projects that they work on here.

I won't go into too many details in terms of examples of the types of projects that are done in this course, but I do have a link to the course website, if you all are interested. One thing that I will mention is that, the important thing to remember here is that in terms of weaving this idea of reproducibility through the curriculum, the tool kit grows along with the complexity of computation. That, I think, helps with students by in. We're not just throwing a bunch of tools at them and saying, "You should learn them all." Instead, what we're saying is, "Here is the amount of data analysis you're expected to do in this course, and here is a set of tools that will make it much easier for you to do it, and do it right, and do it cleanly, and get feedback in the most efficient manner, and also be able to collaborate in the most efficient manner."

So, what comes next. In terms of the what, what we would like to do in our curriculum is to continue weaving this idea through, and introduce in in our Capstone course, or the Senior Thesis and Independent Study. How do we do this? First of all, we're going to need instructor buy in, because it's costly, especially when it comes to an independent study or a senior thesis, a variety of faculty members from the department might be advising the students who are working on these. Most certainly, these faculty do not necessarily want additional checking, on their part, added on to their workload. One way of doing that is to make this a part of the assessment, so that the students recognize the importance of doing their data analysis reproducible, and are rewarded for it as part of their assessment.

Another way is to work with an easily adoptable framework. Something to say to either students or their faculty, saying, "From now on, your projects must be reproducible." That's not sufficient. What we really need is also, a framework where both the faculty who are advising this work, and the students who are working on it, can easily adopt.

One example for this is the steps to reproducible research that Karl mentioned earlier. Project TIER also has a nice framework, that uses the open science framework, underlying that, a project here Ben mentioned at the beginning at the talk. I would especially encourage people who might be working not with R, but maybe perhaps something with like Stata, to take a look at the work done by Project TIER as well, as a protocol that they have developed. Works for Stata as well. And, you might want to work at the reproducible science curriculum. If you click on these, on the slides these should be live links that will take you to the relevant web pages here, and the Reproducible Science Curriculum is a two day workshop for scientists to do their data analysis reproducibly and there are nice steps outlined there, that one might be able to grab and adopt for their own use, and then develop this framework that might be useful for their own curriculum.

In terms of our tool kit, I mentioned R a lot, and the reason why I have used R, beyond the fact that I am an R user in terms of my applied statistic work, is that the built-in seamless ecosystem with RStudio, really makes things like literate programming, and version control Git and GitHub, a lot easier for the students. So, even though they are learning new syntax, they are using additional tools, it all looks like it's happening in one environment for them. That is very useful for at the very beginning, in terms of getting started. And we'll take them through a majority of their work. Is it possible to do this with other languages? Sure. With any scripting language this is possible. Just keep in mind that there may be more overhead in some, than others.

Let's lastly talk about the side effects for students and instructors. What do I mean by side effects? What I mean here is, obviously, there's the goal of teaching data analysis reproducibly, and that's that your students are learning to do better science. But beyond that, for instructors, it makes question and answer easier, because students are doing their work in this single R Markdown document. They can, if they have a question for you, they can simply send you that document or a relevant snip-it from it, so you never really have to worry about answering questions like, "But in my computer, when I run this, this happens." It allows for the environment to be as similar as possible, between the instructor and the student, and it makes Q&A easier. It also makes grading a lot easier. One thing that we've done in our, especially in the lower level courses, is instead of letting the students start with MPR Markdown documents, we provide templates for them for each data analysis lab, so everything is organized exactly the same way. You're not having to sort through, perhaps, a disorganized document. They are simply entering in their code and narrative into the spots that are pre-designed in the template, and that does certainly make grading easier, more efficient and effective.

For students, it really helps collaboration. Instead of sending each other code pieces in email, and potentially being in different spots in terms of data analysis on their individual computers, the fact that they have this single R Markdown document that they can share, means everybody is always at the same spot, in terms of the teamwork. Additionally, if they are using a tool like GitHub, they can visually see any differences between what they had committed, and what fellow teammates may have done, after they had done the commit. So, these tools really make life easier for the students for collaboration, and additionally for self promotion. It's pretty nice to be a first year

undergraduate who on their resume is able to say something like, "Yes, I know R. I can generate reports in R Markdown. And I am familiar with Git and GitHub, and I've worked with those before."

I think that's all that I have prepared today. The slides are here. If you have questions for me, don't hesitate to contact me either on Twitter or my email address. In the repository for the talk, you will also see links to some of the resources and course pages. So if these courses, or the work flows that we've adopted in these courses seem like something you want to adopt in your own courses, take a look at the course pages. Hopefully there's sufficient information there to get you started. And feel free to reach out to me, as well.

Ben: Great! Thank you Mine! I guess, I don't know. We should all be clapping somewhere in our respective offices, or wherever we are. We have a bunch of good questions that have come in. We're a little bit over budget on time, but I believe that we have the ability to continue talking for as long as people want to hang on the line. There's a question here about suggestions for the best way to compile results in tables. So, the person says, "I use R Markdown and knitr, but still find making publishable quality tables challenging. Which R packages and functions do you suggest for reproducible research tables?" Karl or Mine, do you want to weigh in on that?

Karl: I also find publication quality tables to be painful. For my own papers, I will write them in LaTeX, and I'll use the xtable package to make the table. But, I guess, most of the time, the tables that I make in informal analysis reports, I won't even try to make them look that pretty. I will just organize the data in a tabular form and just print it out, and you know, rounding things appropriately. Like I said, the html based tables, either with the kable function that's part of knitr, or with xtable, I haven't been very happy with.

Ben: Yeah, I tend - Go ahead.

Karl: I tend to brute force it. You know, and you know, if you're trying to get a table into word, I've not found a very nice way.

Ben: I tend to agree. I'm finishing a book right now, and we did write the book in LaTeX, compiled with knitr, and there's a bunch of tables. The xtable function can do a lot of things, but you do have to dig a bit if you want to do something non trivial in there.

Mine: Another option is also to save some of the results as additional objects that you then manually put into certain columns of the table. It is not necessarily using a particular package's functionality, but I've heard that that can sometimes allow for the highest level of customization of the table.

Ben: Okay, we have another question about version control, and the commenter says that works best on text only files. So, you kind of have to use Markdown or LaTeX for the narrative, what happens if you have collaborators who don't know how to do anything but Microsoft Word?

Karl: Most of my collaborators are that form, so that the scientific paper will end up being in word, and not reproducible. My approach is to make a side-by-side, R Markdown based, reproducible report, that includes all the results that are going to appear in the actual paper. So you can lay them side by side and make sure that the things that ended up in word were exactly what came out of my analysis. Or, you know, my collaborator might send me a list of further questions, for me to look at, I'll make an R Markdown based analysis report that says, "Here's the question, here's my answer. Here's a figure. Here's the question, here's my answer. Here's a figure." So, I do it by just having a separate report that you can compare with the non-reproducible word document that the collaborator makes.

Ben: I think that is a distinction worth making, that you know, in a lot of ways, Microsoft Word is itself, sort of fundamentally not reproducible, because it can't be scripted. Markdown does have the ability to render to word, but then if you make changes in word, there's no way to go back.

Next question is about, "Karl talked about Make, and how that can help you link together multiple files. What if my analysis requires several separate scripts? Can this be implemented in R Markdown?"

Karl: Yes, it certainly can. You could use the source command, to run a script in a separate file, you know, call that within a code chunk in the R Markdown document. The key advantage of Make, I think would be that, it will rerun only the pieces that you need, and it's maybe somewhat more transparent what the dependencies are. But, you can skip Make and have your R Markdown document control the whole process if you want.

Ben: Okay. We've got a question for Mine. "What are the biggest hurdles to implementing an R or R Markdown based course? Especially for non-stat majors?"

Mine: I think that the biggest hurdle, I would say, would be differences in the set ups of the students. So, if students are using R on their own computer, even if they follow the exact same steps for installation of R, RStudio, and installation of the packages, it is possible that things will still be buggy, and it might be very difficult to help them out, if you're not familiar with their operating system. One way that we've gotten around that is by running an RStudio server, so that students are always logging on to the same environment. I would very strongly recommend, especially if you are teaching a group of students where any frustration caused by such bugs, or such unexpected behavior on their computer, where you're running codes that you're getting something, and they're running codes and they're getting something else, if that could create a level of frustration that they may not be comfortable with, going with the server approach makes things a lot cleaner. That way, the packages that are installed are all the same version, so the rendered reports look exactly the same.

Another potential hurdle could be the fact that is additional syntax to use R Markdown, but I have found that in comparison to the first hurdle that I described, that's close to nothing. By nature, Markdown is a very simple language, and especially if you are providing templates that have a lot of hand holding in them, at least to get them

started, it's become very easy to overcome that hurdle. In my course, for example, we have a set of ten or so labs throughout the semester, and the first lab, the template we provide, even has answers for some of the questions. So they see exactly what's expected of them, and where things go in terms of within a code chunk, because it's code or outside of a code chunk, because that's narrative. As the semester progresses, those templates get slimmer and slimmer, and by the end of the semester the students start with nothing and are able to produce their own data analysis project.

Ben: Are you able to comment on the shared project feature in RStudio server?

Mine: Yeah. I have not actually used that myself, because of how we have set up our servers. The way they are set up, it doesn't allow for that, but that has to do with our setup. RStudio server also has this new capability for shared projects, where it is very similar to a Google Doc environment, and students are, in this day and age, very familiar and incredibly comfortable, I think, with Google Docs. So, as you are making edits, somebody who is in the same shared directory is able to see your work. There is a little bit of instructional overhead here, because you need to set up the shared project such that students who are working in the same team are able to access the same project. But once that set up is completed, it works simply like Google Docs works. So, really, they don't even send some R Markdown file around. They're able to log on and do the data analysis on the same environment.

Ben: I think our last question is about, "What do you do if you want to achieve a reproducible research project, but the data itself is either something that can't be public, or potentially even can't be accessed after the initial work has been done?" Do you have any thoughts on how you might do that?

Mine: Well, in terms of the data- Oh, go ahead Karl.

Karl: If the data can't be seen, it would be hard, I mean, it would be impossible for anyone else to reproduce it. I guess my interpretation of the question looks more like, a project was accomplished using old standard practice, later after the thing has been published you want to go back and turn it into a reproducible product, tips for doing that. My answer to that version of the question would be that you just have to set aside two days, or a week, to go and sort of move over one directory, and start over making use of the code that you'd written before, and try to make an organized version of the full analysis that gets you to the same end point, but in a way that someone else can reproduce.

Mine: I guess one comment I have, which we did mention this earlier, is that the fact that the data can not be shared publicly, that does mean you don't want that data to live on GitHub, but it does not necessarily mean that you can't do local version control through Git. Obviously, that does mean you're losing some of the nice visual capability of using GitHub. I do feel like working with private data, because things can't be public, and because the reproducibility, kind of the idea of reproducibility and the idea of open science tend to go hand and hand, it does seem to make people think, "Well, I have private data, so I just won't do any of this." I feel like that's not the right attitude. I think

the way we want to think about it is, "I have private data. There are some limitations around what I can do and where I can put it, but what of these best practices can I still use, and just not make the data bit public?" So, yes. Someone can't just download that whole repository folder and get to the same answers, but perhaps your code is okay to share. Perhaps the functions that you've developed are okay to share. Or perhaps none of it is okay to share publicly, but it's okay to share within your lab group, or your collaborators, or just with yourself.

Karl: Yeah exactly. And maybe there's a, you know, set of summary statistics from the data that you can at least show what you're doing from that point on.

Mine: Yeah.

Karl: [crosstalk 01:11:05]

Ben: Okay, I have- Sorry. Go ahead, Karl.

Karl: Another experience that I had was that, I tried to make an analysis available to others, and realized that, at that point, the way in which others would be able to access the data was in a quite different form than the way I'd been storing it locally. So, I had to redo a bunch of analysis scripts to make use of the public version, of what the data would be. How it would be stored. Which is kind of a big pain. I guess I've come to learn that I should think about the way in which the data will sit when others will have access to it, and have all my scripts start from that, rather than what I viewed to be the most convenient form.

Ben: Totally. Okay, I have one final question, that comes from me. We've talked a lot about how to do reproducible research, and the kind of tools that are helpful for doing that, what are the missing pieces? Are there any missing pieces? And if so, what are they and, you know, what is your wishlist for things that you wish you could do, but maybe can't? Besides tables in LaTeX.

Mine: I think that one thing that comes to mind, is something you mentioned, which is, it's possible to write out to a Word documents, but you can't, if a collaborator makes edits on that, you can't really go back. Now, I have no idea how, technically, that would be possible, but something that allows for this collaboration seamlessly, with people who are not familiar with the tool kits that we're describing that make data analysis a lot easier. But, if all you're doing is working on the narrative of a paper, perhaps you're not necessarily interested in learning Markdown. So, something that makes that collaboration a bit easier would be helpful, because I think the fact that that step is missing, also turns people off from getting started in a reproducible fashion, from the beginning.

Ben: Yeah. One feature that I do like about Google Docs or Word is the comments feature.

Mine: Yeah.

Ben: Which I guess you can do through GitHub, but not really like within a Markdown Document itself.

Karl: For me, the two things that I want are training and time. All of this stuff takes a lot of time, and that, even now, training to run these new tools is not widely available.

Ben: Well, if you figure out a way to get more time, please let me know, because I could use some. All right, and I think we will stop there. So, let me thank Karl and Mine again, for two great talks! And I hope you enjoyed this webinar. I believe this is going to be archived and available on the web. So, I hope you enjoyed the talk, and we'll see you soon!

Karl: Thank you!

Mine: Thank you!