A framework for mentored data science research

Abstract

We design a mentoring framework to guide undergraduate researchers through individualized research projects in data science. Our framework involves research question formulation, data acquisition, data analysis and visualization, and presentation and communication of results. Our two honors students, whose projects serve as case studies for our framework, completed all components of the individualized research projects. We found that data science research skills, self-confidence in research ability, and professional interest in data science increased for both students. We describe our successes, lessons learned, and ideas for others to build similar frameworks.

Introduction

The need to analyze unprecedentedly large volumes of information combined with the development of faster and more powerful computers has fueled advances in data science methods for big data. Similar causes have led to a need for greater numbers of scientists with quantitative skills. In efforts to enhance training and mentoring for students, we created a program that emphasizes many transferable skills that contribute to career success in data science.

We elected to work with social media data. In making this decision, we recognized that social media data, such as tweets from Twitter, can be acquired with little cost and that there is growing research interest in social media in many social science disciplines, including political science, communication studies, and sociology. We also anticipated that our undergraduate trainees might be intrigued by the possibility of analyzing social media data, since many young adults use Facebook, Twitter, Instagram, and related sites.

Some social media data, including tweets from Twitter, are available through website application product interfaces (APIs). Twitter shares, via a streaming API, a sample of approximately one percent of all tweets during an API query time period ("Sampled Stream," n.d.). Researchers have studied tweets for a variety of purposes, including inference of relationships and social networks among users (Lin et al. 2011); determination of authorship of specific tweets when multiple persons share a single account (Robinson 2016); and study of rhetoric in recruiting political supporters (Pelled et al. 2018; Wells et al. 2016). Recognizing the potential utility of tweets for data science research and teaching, we created a collection of tweets over time by repeated querying of the Twitter streaming API.

Nolan and Temple Lang (2010) argue for students to work with real data. Working with real data allows students to develop skill not only in statistical analysis, but also in data transfer from online sources, in data storage, and in using data from multiple file formats. In the case of Twitter data, tweets are stored in Javascript Object Notation (JSON) ("Consuming Streaming Data," n.d.; "Introducing Json," n.d.).

Mentoring in the work place and in higher education can have many benefits, including improving students' development as thinkers and scholars, confidence in their own abilities, integration into the campus community, and interest in graduate training (Baker and Griffin 2010; Higgins and Kram 2001). A key component of our data science mentoring framework is the emphasis on using real data to answer scientific questions. We believe that this process develops problem-solving skills that students will need in their future careers in data science. We encouraged students to articulate a scientific research question, translate that question into quantitative and statistical terms, determine which data could be used to address the question, acquire the data, analyze data, visualize results, and communicate what they learned.

Our backgrounds

During the time when we first implemented our framework, we served as early-career instructors in the statistics department at the University of Wisconsin-Madison. One of us had prior experience in mentoring students, while the other had none. Our initial conceptualization of mentoring drew heavily on ideas we first encountered in professional development activities, including the Delta Program's mentoring class (https://delta.wisc.edu) and Handelsman et al. (2005). Professor Erik Nordheim's class, "Teaching Statistics", influenced our approach to and philosophy of teaching statistics. We studied with Professor Nordheim early in our teaching careers, and his course's emphasis on backward design and active learning continues to influence our teaching practices (https://delta.wisc.edu).

We both have experience in teaching undergraduate introductory statistics courses with enrollments of more than 100 students. Through our interactions with students in these classes, we've grown to value not only the ideas in a traditional introductory course, but also the need to prepare students with the essential skills needed for success in data science. Nolan and Temple Lang (2010) summarizes these skill sets in the following three goals:

1. broaden statistical computing to include emerging areas
2. deepen computational reasoning skills

3. combine computational topics with data analysis in the practice of statistics

To these three praiseworthy goals, we add a fourth:

4. develop skills in reproducible research to promote open science practices

We see the fourth goal as an equal with the first three from Nolan and Temple Lang (2010). Data scientists are uniquely positioned to promote open science practices, including the free sharing of data, code, and instructions for their use. The need for science to be more transparent and more reproducible elevate this goal to the level of the first three.

Below, we detail our methods for creating a reproducible framework for undergraduate data science research. We describe our results before concluding with lessons learned, things we could have done differently, and recommendations for future mentors who may use and extend our framework.

Methods

We designed and implemented a framework for mentored undergraduate data science research projects with big data. Below, we describe our initial framework and connect it to ideas from Nolan and Temple Lang (2010).

Framework Overview

t

step	details
Question formulation	List ideas
Question formulation	Prioritize ideas
Question formulation	Check data availability
Data acquisition	Download tweets and other data
Data acquisition	Parse tweet JSON
Data acquisition	Organize data as times series
Data analysis and visualization	Exploratory plots
Data analysis and visualization	Summary statistics
Presentation and communication	Interpret and summarize graphs and summary statistics
Presentation and communication	Prepare written materials and figures
Presentation and communication	Share findings as posters, slides, and reports

Framework implementation

Research question formulation

Our mentored research framework begins with brainstorming scientific research ideas based on the student's interests. This enables us to craft a project that excites the student. With the results of brainstorming sessions, we (mentors and student together) formulate the most promising ideas into scientific hypotheses.

For the most appealing scientific hypotheses, we encourage the student to translate the scientific question into a statistical question that may be addressed with data. This is a crucial step in data science research question formulation. Skill in translating in both directions between scientific and statistical questions is a key communication skill that data science researchers offer.

Data acquisition

We also incorporated data availability into our question formulation. We limited questions to those that could be studied with publicly available data. This practice also enabled reproducibility of our analyses, since students could share the URL from which they accessed data.

Our computational system for acquiring tweets involved several steps. We interacted with the API via the R package `twitteR` (Gentry 2015). We used the free Twitter streaming API that gave us access to approximately one percent stream of all tweets during the specified query time period. To ensure that we collected tweets continuously, we used the linux tool `crontab` to execute our R script every five minutes. Each execution of the R script performed a single streaming API query for five minutes. Twitter's streaming API, at the time of our data collection, enforced rate limits on the frequency and duration of queries. With the above settings, we continuously collected tweets.

We encouraged students to complement tweets with additional data from publicly available sources.

Data analysis and visualization

After identifying research questions and publicly available data, the next step is to decide on informative data visualizations and quantitative analyses. Because both projects involved exploratory analyses of times series, we encouraged students to think about visualizations that might reveal relationships over time.

In the case of the event detection project, our student created word clouds for every inferred "topic". He also presented most probable words from each inferred topic, i.e., each distribution over words, as a bar plot.

The student working on sentiment analysis and market index prices plotted a daily sentiment "score" over time and presented it beside a plot of daily market index prices and compared the two plots.

Presentation and communication of results

Students presented their research in a variety of settings. Each student presented at the annual undergraduate statistics poster session. We also encouraged them to present at the annual university-wide undergraduate research symposium.

In planning with students for poster and slide presentations, we emphasized the importance of succinctly stating the research question and its scientific context. After clarifying the importance of the question, the student could proceed with explaining many of the elements that we've described above. Namely, the student would discuss the analyzed data and its acquisition while noting any shortcomings or biases of the data. For oral presentations, we suggested that students cautiously limit discussion of statistical methods, with the caveat that they prepare to answer detailed methodological inquiries during the question and answer session. Our students created powerful data visualizations for their projects. Their presentations also included their major results and future research directions.

In efforts to develop student written communication, we encouraged both students to prepare a written senior thesis document that detailed their research. In the senior thesis, we suggested that the students describe in rigorous detail their statistical methods. The rationale for this distinction, relative to the oral presentation, is that a reader doesn't have access to a question and answer session, while a poster session attendee may freely ask questions of the author.

Examples

Examples may help to demonstrate our approach to identifying a statistical research question. One of our students had interests in acquiring and using social media posts. We helped her in brainstorming ideas for research involving social media sources like Facebook and Twitter. Through this brainstorming, we recognized that she had a parallel interest in financial markets. Our student hypothesized that sentiment analysis of finance-related tweets might reflect trends in financial market index prices. On days when the market index prices increase, sentiment analysis of finance-related tweets might reveal more use of positive words, while days with decreasing prices might have more negative words in finance-related tweets.

A second student wanted to study tweets over time and entertainment events that garner lots of attention in social media. We encouraged this student to develop a strategy for event detection from tweets over time. The rationale is that a big entertainment event, such as the National Football League's Super Bowl game, might generate enough tweets that Super Bowl-related words would appear with high weights in results from latent Dirichlet allocation modeling of collections of tweets at distinct time points. We reasoned that Super Bowl-related topics might appear during the Super Bowl and vanish soon after the game's conclusion.

Broaden statistical computing to include emerging areas

Our framework broadens statistical computing by including the emerging areas of social media data analysis, sentiment analysis, and topic modeling. Both students used Twitter tweets, which we accessed through a Twitter streaming API. One student chose to apply topic modeling to tweets. The other elected to apply sentiment analysis to daily tweet collections. By using these methods, sentiment analysis and topic modeling, students learned an aspect of text analysis that is not taught in traditional statistics courses.

Deepen computational reasoning skills

Our framework encourages students to deepen computational reasoning skills in several ways. First, they work with a variety of internet-based data to answer research questions. In the two example cases, our students collected tweets over time and gathered complementary data from other resources, including daily closing prices of stock market indexes. This gave students opportunities to think creatively about what data to acquire and how to use multiple data sources in a single cohesive project.

Second, the students worked with a variety of data structures. The Twitter streaming API returns tweets as JSON (Javascript Object Notation). Because distinct Twitter users may provide different pieces of profile information, there is variability in the structure of each tweet's JSON. Additionally, tweet metadata fields may appear in any order (https://developer.twitter.com/en/docs/tutorials/consuming-streaming-data). Students needed to recognize this and to write code that accommodated these variations in tweet data structure. Additional variability in tweet structure arose due to changes in the API. The evolving nature of JSON tweet structure (https://developer.twitter.com/en/docs/tweets/data-dictionary/guides/tweet-timeline) required students to write flexible code that could incorporate newly introduced or deprecated metadata.

Students wrote R code to parse and organize tweet JSON. They organized their R code into a package, and shared it on Github (https://github.com/rturn/parseTweetFiles). Each tweet's JSON included required fields, and, possibly, some optional fields. Thus, students' code needed to accommodate variability in tweet structure.

Combine computational topics with data analysis in the practice of statistics

Both mentored students combined computing with data analysis in the practice of statistics. They used a combination of latent dirichlet allocation topic modeling, sentiment analysis, and time series analysis to reach conclusions about real world phenomena.

Both drew heavily on the collection of tweets. One student examined Standard and Poor's 500 index daily closing prices over time. She also analyzed sentiments from each day's stock market-related tweets to look for relationships between tweet sentiment and stock market prices.

Our other student focused on developing detection methods for social media events through topic modeling of tweets at different time periods. As a proof of principle, he fitted topic models to collections of tweets preceding, during, and following the National Football League's Super Bowl game. He hypothesized that topics would evolve over time, with football-related tweets appearing during the football game and disappearing soon after conclusion of the game.

Both students analyzed tweets as texts. This first required them to write code to parse the JSON that the API returns. Once they had isolated the tweet text from its metadata, they parsed the tweet text into words for use in sentiment analysis and topic modeling. For the stock market project, they analyzed only those tweets that contained finance-related keywords. Sentiment analysis involved comparisons of tweet words to a dictionary that mapped words to sentiments. This yielded a net sentiment score for each tweet. They then treated tweet sentiment scores as a time series and compared them with daily stock market index closing prices.

The second student project involved latent Dirichlet allocation modeling of tweet words at distinct time points to detect social media events (Blei, Ng, and Jordan 2003). Latent Dirichlet allocation is a bayesian nonparametric method for modeling text corpora as the result of words chosen from topics. The student treated tweets as "documents" (in the parlance of topic modeling). The goal of topic modeling, then, was to infer the underlying "topics" (or probability distributions over words) from a collection of tweets.

Develop and practice skills in reproducible research to promote open science

With the goal of promoting transparency in our research, we encouraged students to use `git` for version control of their code and documents and to share their code via the website Github (https://github.com). One student also enrolled in Karl Broman's class on tools for reproducible research (https://kbroman.org/Tools4RR/). This class features `git` and Github throughout its lectures and activities.

As we stated above, the students created an R package, `parseTweetFiles`, version controlled it with `git`, and shared it via Github.

Results

We applied the project framework to our mentoring of two students. Both engaged in 12 months of research during their senior year of undergraduate studies in statistics. Below, we describe three categories of outcomes:

1. student outcomes
2. mentor outcomes
3. scholarly outcomes

Student Outcomes

We subjectively assessed student outcomes through conversations in our weekly student research meetings. Both students showed increases in confidence and ability to do data science research.

Both students secured positions in data science after graduation. One student enrolled in a statistics graduate program, while the other pursued employment in health care analytics.

Students benefited from our emphasis on the four central concepts, three from Nolan and Temple Lang (2010) plus reproducible research skills. The research projects successfully drew on emerging areas of statistical computing, namely text analysis. They combined computational topics, including topic modeling and time series methods, with data analysis in the practice of statistics. Although we didn't formally measure them, our informal assessment indicates that students' computational reasoning skills increased over the duration of our projects. Students used a variety of computing tools and methods to arrive at a practical solution to a selected task. They became more skilled in computing with R and shell scripts and more fluid in their verbal explanations during our regular meetings (R Core Team 2019).

Our framework's emphasis on reproducible research skills is evidenced by the students' R package, `parseTweetFiles`, which is both version controlled with `git` and shared via Github.com.

Mentor outcomes

We grew as mentors during our work with the two students. We successfully guided junior scientists through a productive, hands-on research experience, and we anticipate refining the framework in future iterations.

Scholarly outcomes

Our scholarly contributions include the `parseTweetFiles` R package on Github (https://github.com/rturn/parseTweetFiles) and presentations at conferences such as useR! 2016 (R users' conference) and local poster sessions. Additionally, both students prepared end-of-project reports on their research.

Discussion

Benefits of our framework

The student test cases for our framework demonstrated greater self-confidence and greater proficiency in data science skills over the course of the research projects. They used real-world data sources to address real scientific research questions. Additionally, they showed great interest in quantitative and data science careers. After graduation, one student immediately enrolled in statistics graduate training, while the other sought employment in health care analytics.

Critiques of our framework

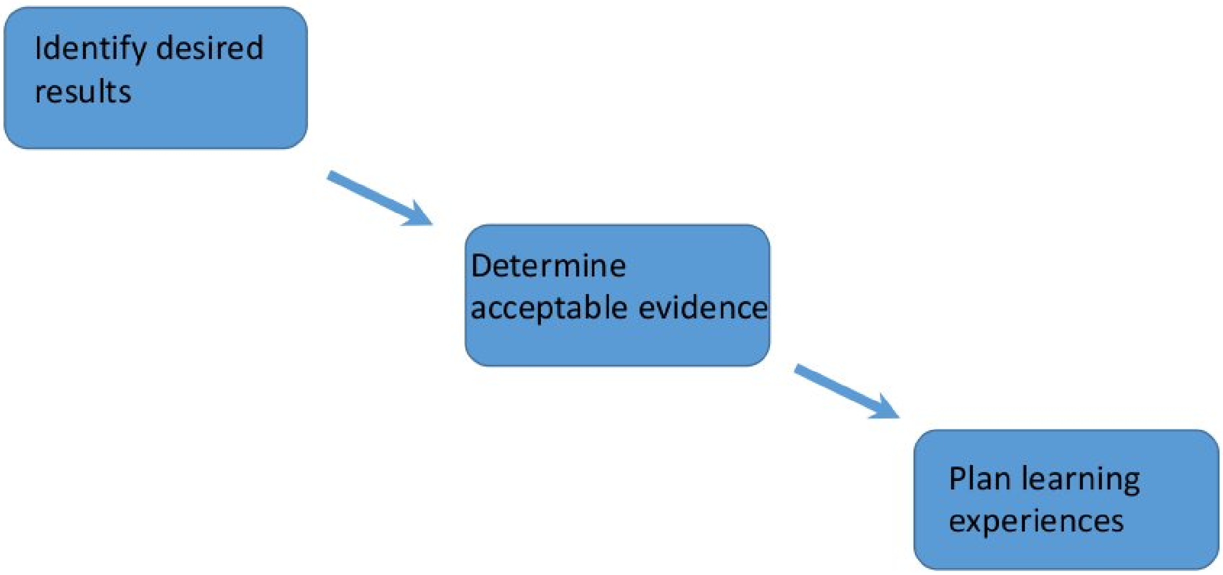
From our current perspective, we offer a number of framework critiques and opportunities for improvement. Our measure of students' self-confidence in research ability was merely subjective. In future iterations of our framework, we would like to measure systematic and objective outcomes. One strategy for implementing this is to administer a survey, including questions from Vance et al. (2017), both before and after the mentored research project. We would use survey questions that focused on student beliefs about themselves, their skills, and their future careers.

One shortcoming of our initial framework was the relative lack of emphasis on best practices for computational reproducibility. This is one area that we would like to rectify in future mentoring activities. The University of Wisconsin-Madison has periodically offered a semester course in best practices for computationally reproducible research (https://kbroman.org/Tools4RR/). We especially see collaborative version control systems, such as Git and Github, as essential tools for the modern data scientist.

1. assessment of data science skills
2. assessment of attitudes (pre and post survey??)

Framework development with backward design

In future research, we will continue to develop our framework for undergraduate data science research by explicitly incorporating backward design principles (Wiggins and McTighe 2005). Following Wiggins and McTighe (2005), we will identify desired results, determined acceptable evidence, and planned learning experiences.



Before identifying desired results, we will prioritize topics from Nolan and Temple Lang (2010). Specifically, we will assign all terms from Figure 1 of Nolan and Temple Lang (2010) into one of three categories:

1. worth being familiar with
2. important to know and do
3. enduring understanding

We've tabulated below the Nolan and Temple Lang (2010) terms for the current framework and its student projects.

Prioritizing Key Terms from Figure 1 of @nolan2010computing

xxx

Term	Circle
R packages	Enduring understanding
debugging	Enduring understanding
shell tools	Enduring understanding
reproducible computation	Enduring understanding
text editors	Enduring understanding

version control	Enduring understanding
file system concepts	Enduring understanding
text processing	Enduring understanding
regular expressions	Enduring understanding
EM	Important to know and do
MCMC	Important to know and do
Bayesian computation	Important to know and do
programming scope	Important to know and do
data structures	Important to know and do
portability	Important to know and do
authoring tools	Important to know and do
GUIs	Important to know and do
grammar of graphics	Important to know and do
composition	Important to know and do
linear algebra decompositions	Worth being familiar with
representation of numbers	Worth being familiar with
RNG	Worth being familiar with
optimization	Worth being familiar with
numerical algorithms	Worth being familiar with
efficiency	Worth being familiar with
parallel computing	Worth being familiar with
modeling language	Worth being familiar with
distributed computing	Worth being familiar with
compiled languages	Worth being familiar with
OOP	Worth being familiar with
symbolic math	Worth being familiar with
data bases	Worth being familiar with
I/O	Worth being familiar with
Flash	Worth being familiar with
HTTP	Worth being familiar with
XML	Worth being familiar with
SOAP	Worth being familiar with

SVG	Worth being familiar with
KML	Worth being familiar with
grid	Worth being familiar with
lattice	Worth being familiar with
event programming	Worth being familiar with
maps	Worth being familiar with
interactivity	Worth being familiar with
animation	Worth being familiar with
perception	Worth being familiar with
color	Worth being familiar with
raster/vector graphics	Worth being familiar with

Potential benefits of incorporating backward design ideas include clearer articulation of goals and better assessment of goal achievement.

We see our framework as one contribution to scholarship on improving data science training programs. Given the increasing economic need, in the USA and abroad, for data scientists and other researchers with quantitative training, we anticipate that our framework and its future iterations will continue to prepare students for data science careers by offering training in tangible and transferable analytic skills in the context of solving scientific questions.

Integrating more mentoring activities

Our framework would benefit students more if we explicitly incorporate more mentoring activities. Through professional development courses at the University of Wisconsin-Madison's Delta Program, we received training in how to offer professional support to students. While we both informally supported our students, the Delta Program suggested ways to encourage the student's professional development through structured conversations and goal-setting. Additions like this would only enhance our framework.

Baker and Griffin (2010) discuss the role of faculty "developers" in student success. A faculty "developer", as envisioned by Higgins and Kram (2001), offers not only psychosocial and career support, like a mentor, but also supports students' academic goals. Such relationships between developers and students benefit both parties. The student gets support while the developer refines her teaching and expands her scholarly network. We anticipate expanding our framework to more holistically support students.

References

Baker, Vicki L, and Kimberly A Griffin. 2010. "Beyond Mentoring and Advising: Toward Understanding the Role of Faculty 'Developers' in Student Success." *About Campus* 14 (6). Wiley Online Library: 2–8.

Blei, David M, Andrew Y Ng, and Michael I Jordan. 2003. "Latent Dirichlet Allocation." *Journal of Machine Learning Research* 3 (Jan): 993–1022.

"Consuming Streaming Data." n.d. https://developer.twitter.com/en/docs/tutorials/consuming-streaming-data.

Gentry, Jeff. 2015. "TwitteR: R Based Twitter Client." https://CRAN.R-project.org/package=twitteR.

Handelsman, Jo, Christine Pfund, Sarah Miller Lauffer, and Christine Maidl Pribbenow. 2005. *Entering Mentoring.*

Higgins, Monica C, and Kathy E Kram. 2001. "Reconceptualizing Mentoring at Work: A Developmental Network Perspective." *Academy of Management Review* 26 (2). Academy of Management Briarcliff Manor, NY 10510: 264–88.

"Introducing Json." n.d. https://json.org.

Lin, Cindy Xide, Qiaozhu Mei, Jiawei Han, Yunliang Jiang, and Marina Danilevsky. 2011. "The Joint Inference of Topic Diffusion and Evolution in Social Communities." In *2011 Ieee 11th International Conference on Data Mining*, 378–87. IEEE.

Nolan, Deborah, and Duncan Temple Lang. 2010. "Computing in the Statistics Curricula." *The American Statistician* 64 (2). Taylor & Francis: 97–107.

Pelled, Ayellet, Josephine Lukito, Fred Boehm, JungHwan Yang, and Dhavan Shah. 2018. "'Little Marco,''Lyin'Ted,''Crooked Hillary,' and the 'Biased' Media: How Trump Used Twitter to Attack and Organize." In *Digital Discussions*, 176–96. Routledge.

R Core Team. 2019. *R: A Language and Environment for Statistical Computing.* Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Robinson, David. 2016. "Text Analysis of Trump's Tweets Confirms He Writes Only the (Angrier) Android Half." http://varianceexplained.org/r/trump-tweets/.

"Sampled Stream." n.d. https://developer.twitter.com/en/docs/labs/sampled-stream/overview.

Vance, Eric A, Erin Tanenbaum, Amarjot Kaur, Mark C Otto, and Richard Morris. 2017. "An Eight-Step Guide to Creating and Sustaining a Mentoring Program." *The American Statistician* 71 (1). Taylor & Francis:

23–29.

Wells, Chris, Dhavan V Shah, Jon C Pevehouse, JungHwan Yang, Ayellet Pelled, Frederick Boehm, Josephine Lukito, Shreenita Ghosh, and Jessica L Schmidt. 2016. "How Trump Drove Coverage to the Nomination: Hybrid Media Campaigning." *Political Communication* 33 (4). Taylor & Francis: 669–76.

Wiggins, Grant, and Jay McTighe. 2005. *Understanding by Design.*