What is happening on Twitter? A framework for student research projects with tweets

Frederick J. Boehm and Bret M. Hanlon

1 Abstract

We draw on our experiences with mentoring two students to develop a framework for undergraduate research projects with Twitter data. Leveraging backward design principles, we share our learning objectives and rubric for summative assessments. To illustrate the value of Twitter as a data source, we detail methods for collecting and analyzing tweets. We conclude by emphasizing how Twitter text analysis projects enable students to formulate original research questions, collect and analyze data, and communicate findings and their implications.

2 Introduction

Twitter has profoundly changed how we communicate. In only 280 characters, users instantly contribute to public conversations on politics, current events, sports, media, and many other topics. Recent development of accessible statistical methods for large-scale text analysis now enable instructors to use tweets as contemporary pedagogical tools in guiding undergraduate research projects. We guided two statistics students in their senior research projects. Both students used tweets to address novel research questions. We share products of their research in supplementary files. Because their data are no longer available, we present as a case study one analysis with tweets from May 2020. We share our data and computer code to encourage others to undertake tweet text analysis research. We also describe methods for creating a collection of tweets.

Some social media data, including tweets from Twitter, is available through website application product interfaces (APIs). By way of a streaming API, Twitter shares a sample of approximately one percent of all tweets during an API query time period ("Sampled stream" 2019). Any Twitter user can freely access this one percent sample, whereas access to a larger selection is available to researchers for a fee.

Studies of Twitter conversations have yielded valuable insights into modern culture. Using large collections of tweets, scholars have investigated diverse research questions, including the inference of relationships and social networks among Twitter users (Lin et al. 2011); authorship of specific tweets when multiple persons share a single account (Robinson 2016); and 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. We envisioned this collection as a rich resource for data science research projects. This vision grew into two mentored undergraduate student research projects in the 2015-2016 academic year.

In line with recent calls for students to work with real data (Carver et al. 2016; Nolan and Temple Lang 2010), our collection of tweets has served as a valuable resource in our mentoring of undergraduate data science research. Working with real data allows students to develop proficiency not only in statistical analysis, but also in related data science skills, including data transfer from online sources, data storage, using data from multiple file formats, and communicating findings. Collaboratively asking and addressing novel questions with our collection of tweets gave mentored students opportunities to develop competency in all of these areas.

While our tweet collection enables us to address many possible research questions, the dynamic content of tweets over time particularly piqued our interest. We hypothesized that high-profile social media events would generate a high volume of tweets, and that we would detect social media events through changes in tweet topic content over time. We discuss in detail below one approach to studying this question. In the sections that follow, we detail our backward design-inspired approach to writing learning objectives, preliminary research mentoring considerations, data science methods for collecting and analyzing tweets, analysis results, and ideas on assessment and advanced topics.

3 Structure of mentored research

3.1 Backward design

Backward design principles guided our planning and informed the writing of learning objectives (Wiggins and McTighe 2005). Following Wiggins and McTighe (2005), we began by listing what students, at the end of their thesis research, should be able to do, understand, and know. We then classified each of these items into one of three categories: enduring understanding, important to know and do, and worth being familiar with (Wiggins and McTighe 2005) (Table 1). While other researchers may categorize these skills differently, our assignments reflect our projects' priorities. Nearly all of the skills in Table 1 are transferable. They apply

Table 1: Classifying project skills

Skill	Category
Structure research project files as R package	Worth being familiar with
Communicate results in speaking and in writing	Enduring understanding
Formulate a research question	Enduring understanding
Develop data science strategies to address research question	Enduring understanding
Use Github to share code and documentation	Important to know and do
Use git for version control	Important to know and do
Use text analysis tools to analyze tweets	Enduring understanding
Use cluster computing as needed	Worth being familiar with
Use data visualization to clarify and inform quantitative analyses	Important to know and do
Translate analysis results into scientific conclusions	Enduring understanding
Incorporate supplementary data sources into analysis	Important to know and do
Acquire data from internet sources	Important to know and do
Describe assumptions and limitations of statistical analyses	Enduring understanding

not merely to thesis projects, but also to data science research in general.

3.2 Learning objectives

We translated our prioritized list of skills that students should be able to do, understand, and know into learning objectives (Table 1). We phrased learning objectives in a manner that enabled their subsequent assessment (Table 2) with formative and summative strategies. These were our four learning objectives:

1. Write R code to perform text analysis of large volumes of tweets (R Core Team 2019).
2. Communicate results in a written report and poster presentation.
3. Translate statistical findings into scientific conclusions.
4. Develop data science strategies to address a scientific research question.

3.3 Preliminary research mentoring considerations

We developed research goals with students in a series of discussions. As trainees began their senior research projects, we spoke in detail about both their research interests and goals and their experience with data analysis software.

Student experience with statistical software varies. In our statistics department, most students learn elementary R computing skills through class assignments. Some students, by concentrating in computer science, learn other data analysis software packages, such as Python. Those who do undergraduate statistics research often learn advanced topics in R computing, such as R package assembly, documentation, and testing. Many develop expertise in linux computing and cluster computing, too.

One of our two students had extensive experience in statistical computing. In addition to R computing skills, she also worked in Python and excelled in shell scripting. She first learned Python in computer science courses. Our second student had extensive experience with R from his statistics courses. His background enabled him to write an R package as part of his senior project. To encourage further development of R computing skills in our two students, we guided them towards the online books “R for Data Science” (Wickham and Grolemund 2016) and “Advanced R” (Wickham 2019).

3.4 Student research interests and goals

Student interests vary, and students’ ability to articulate research goals may be limited. An initial brainstorming session may clarify their interests and encourage them to think critically about goals under the time constraints of their academic schedules. Additionally, we anticipate that sharing completed student project reports will guide student thinking about the scope of possible projects (Supplementary files).

We briefly describe the two student projects to give readers a better sense of research possibilities with tweets. Our first student examined relationships over time between stock market index prices and tweet sentiment. For each day in her 12-month study period, she identified stock market-related tweets with a key word search. With the complete texts of stock market-related tweets for each day, she calculated a daily sentiment score and plotted it over time. Her sentiment score reflected presence of emotion-associated terms (*eg.*, “happy”, “sad”, “mad”, “scared”) in tweet texts. Days with more net positive emotion words in the collected tweets received a higher (positive) daily sentiment score, while days with more net negative words received a negative daily sentiment score. For her final project, she presented plots over time of her daily sentiment scores and daily closing prices of the Standard and Poor’s 500 index. She also explored time series analysis methods to quantify relationships between index prices and sentiment scores.

Our second student developed social media event detection methods with topic models. He hypothesized that tweet content changes over time, and that we might detect these changes by comparing inferred tweet topics from distinct time periods. To validate his hypothesis, he examined tweet content before, during, and after the National Football League’s Super Bowl game in 2015. He reasoned that because the Super Bowl is widely discussed on Twitter, we might detect Super Bowl-related topics from tweets sent during the game, but that the football-related topics would be short-lived in the continuous Twitter stream. We discovered evidence to support his strategy, and we ultimately presented our findings at international and local research meetings. We share below a case study based on this approach.

3.5 Time period

Our two statistics students conducted their research projects during the 2015-2016 academic year. We recommend a full academic year for projects of this magnitude, although a summer or one-semester project is possible. Our students presented their findings at the statistics department's undergraduate poster session near the end of the 2015-2016 academic year (Supplementary files). We present below reproducible R code for analyzing data from May 2020. While these are not the same data that our students analyzed in 2015, the methods and code are very similar to that of our second student's project.

4 Case study methods

To illustrate the value of Twitter data, we present below a reproducible case study. It is essentially a reproduction of our second student's project, but at a distinct time period. In it, we aim to detect a social media event by examining topic content over time. We use Latent Dirichlet Allocation (LDA) models (Blei et al. 2003) to infer topics on three consecutive days centered on Memorial Day 2020. We chose this example case study, instead of the student projects, because of limited data availability for the student projects. Despite this, the case study illustrates the strategy and methods for one of the student projects.

4.1 Case study design

We sought to validate our hypothesis that we could detect a social media event by examining tweet topic content at distinct time periods. As a proof of principle of our event detection strategy, we analyzed tweets before, during, and after Memorial Day (May 25, 2020). We fitted latent Dirichlet allocation models for each of three distinct five-minute periods. The first period began at noon Eastern time on May 24, 2020. Subsequent time periods started 24 and 48 hours later. We defined each time period to be a single collection, or corpus, of tweets.

4.2 Collecting tweets over time

We include here instructions for creating a tweet collection. First, we created a new account on Twitter. With these user credentials, we used the R package `rtweet` to query the API. We used the linux `crontab` software to repeatedly execute R code to submit API queries. Each query lasted five minutes and produced a text file of approximately 130 MB. We timed the API queries so that there was no time lag between queries. We stored tweets resulting from API queries in their native JSON format.

Setting up the query task with `crontab` is straightforward. On our computer, with Ubuntu 20.04 linux operating system, we opened a terminal and typed `crontab -e`. This opened a text file containing user-specified tasks. We added the following line to the bottom of the file before saving and closing the text file.

```
*/5 * * * * R -e 'rtweet::stream_tweets(timeout = (60 * 5),
parse = FALSE, file_name = paste0("~/work/mentoring/mentoring-framework/data/",
lubridate::now(), "-tweets"))'
```

Readers may need to slightly amend the above line to conform to requirements of their operating system's `crontab`. Readers who use Mac OS may proceed as we did, while those with Windows operating systems may consider using the R package "taskscheduleR" to schedule API queries via the Windows task scheduler (Wijffels and Belmans 2018).

4.3 Querying Twitter API to get complete tweets

Twitter API use agreements forbid users from sharing complete API query results. However, Twitter permits users to share tweet identification numbers. With a tweet identification number, a user may query a Twitter API to obtain complete tweet data. In our experience, this process is incomplete; that is, many tweet identification numbers submitted to the Twitter API return no data. Additionally, on repeated querying of the API, different sets of tweet identification numbers return data. This complicates our goal of making all analyses computationally reproducible and motivates our decision to share the tweet IDs of those tweets that we actually analyzed (Supplementary files). Should a reader wish to reproduce our analysis, we anticipate that she will get complete tweet data for all or most of these tweet identification numbers from the API. We provide R code for this task in the supplementary files.

4.4 Tweet structure

Tweets are available from the Twitter API as Javascript Object Notation (JSON) objects ("Introducing JSON" 2020). Every tweet consists of multiple key-value pairs. The number of fields per tweet depends on user settings, retweet status, and other factors ("Introduction to Tweet JSON" 2020). The 31 tweet key-value pairs belong to 12 distinct classes (Supplementary files). The classes are either vectors - numeric, logical, or character - or arrays assembled from the vector classes.

Below is an example of JSON for one tweet.

```
{
  "created_at": "Thu Apr 06 15:24:15 +0000 2017",
```

```
  "id_str": "850006245121695744",
  "text": "1\/ Today we\u2019re sharing our vision for the future of the Twitter API platform!",
  "user": {
    "id": 2244994945,
    "name": "Twitter Dev",
    "screen_name": "TwitterDev",
    "location": "Internet",
    "url": "https:\/\/dev.twitter.com\/",
    "description": "Your official source for Twitter Platform news, updates & events.
    Need technical help? Visit https:\/\/twittercommunity.com\/ \u2328\ufe0f
    #TapIntoTwitter"
  },
  "place": {
  },
  "entities": {
    "hashtags": [
    ],
    "urls": [
      {
        "url": "https:\/\/t.co\/XweGngmxlP",
        "unwound": {
          "url": "https:\/\/cards.twitter.com\/cards\/18ce53wgo4h\/3xo1c",
          "title": "Building the Future of the Twitter API Platform"
        }
      }
    ],
    "user_mentions": [
    ]
  }
}
```


Our analyses use three fields from each tweet: date (“created_at”), tweet identifier (“id_str”), and tweet text (“text”). The “created_at” field is a character string containing the date and time of the tweet. Every tweet has a unique identifier, the "id_str" value. The "text" field contains the unicode representation of the message. After creating a text file with tweet JSON, our next step involved reading and parsing tweets with the R packages `rtweet` (Kearney 2019) and `tidytext` (Silge and Robinson 2016).

4.5 Parsing tweet text

The next task is to wrangle the tweet JSON into a data structure suitable for latent Dirichlet allocation modeling. We used functions from the `rtweet` R package to parse tweet JSON into a data frame. We then divided tweet text into words with functions from the `tidytext` R package. We discarded commonly used "stop words" and emojis.

Latent Dirichlet allocation model fitting requires that the corpus be organized as a document by term matrix. In a document by term matrix, each row corresponds to a single document (a single tweet), and each column is a single term (or word). Each cell contains a count (the number of occurrences of a term in the specified document). We created a document by term matrix with the R function `cast_dtm` from the `tidytext` package.

4.6 Latent Dirichlet allocation

Latent Dirichlet allocation is a statistical method for inferring latent (unobservable) topics (or themes) from a large corpus (or collection) of documents (Blei et al. 2003). We pretend that there's an imaginary process for creating documents in the corpus. For each document, we choose a discrete distribution over topics. For example, some tweets from Memorial Day may refer to the holiday. This may constitute one topic in the corpus. Having chosen a distribution over topics, we then select document words by first drawing a topic from the distribution over topics, then drawing a word from the chosen topic. The goal for latent Dirichlet allocation is to infer both the distribution over topics and the topics (Blei et al. 2003). A topic, in this setting, is a distribution over the vocabulary (the collection of all words in a corpus).

Inference for latent Dirichlet allocation models is performed by either sampling from the posterior distribution or through variational methods. Researchers have devised a variety of Gibbs sampling techniques for these models (Porteous et al. 2008). Variational methods, while using approximations to the posterior distribution, offer the advantage of computational speed (Blei et al. 2017). We used variational methods below.

5 Case study results

We identified the top ten most probable terms for each of ten topics in our models (Figures 1, 2, 3). We plotted the within-topic word probabilities as bar graphs. We see that topic-specific word probabilities seldom exceed 0.05. We also note that some words are heavily weighted in multiple topics. This observation complicates semantic topic interpretation.

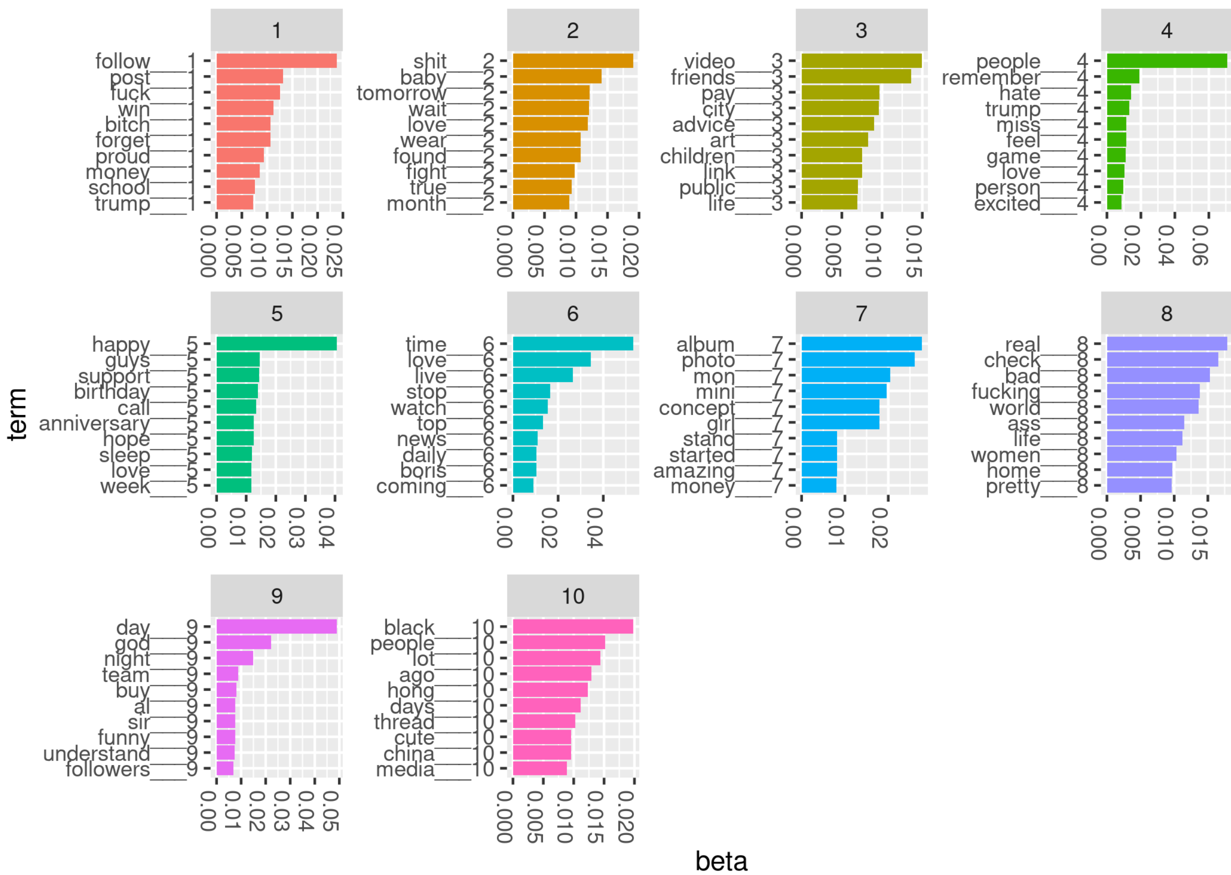

Figure 1: Top terms for LDA model from May 24, 2020

Assigning meaning to topics is an active research area (Chang et al. 2009). Since our interest is in the transient appearance of a new topic, we don't attempt to assign meaning to every topic in our models. We see that topic 7 from May 25 has several words that suggest Memorial Day: memorial, remember, honor, country. A similar topic is not seen on May 24 or May 26. Some topics persist, with distinct word probabilities, across the three days. For example, we see that President Trump features prominently in all three models. We also note, on May 26, topic 10, which reflects discussion of the Amy Cooper Central Park incident (https://www.nytimes.com/2020/05/26/nyregion/amy-cooper-dog-central-park.html).

We also note that the murder of George Floyd occurred on May 25. Our last examined time period, from

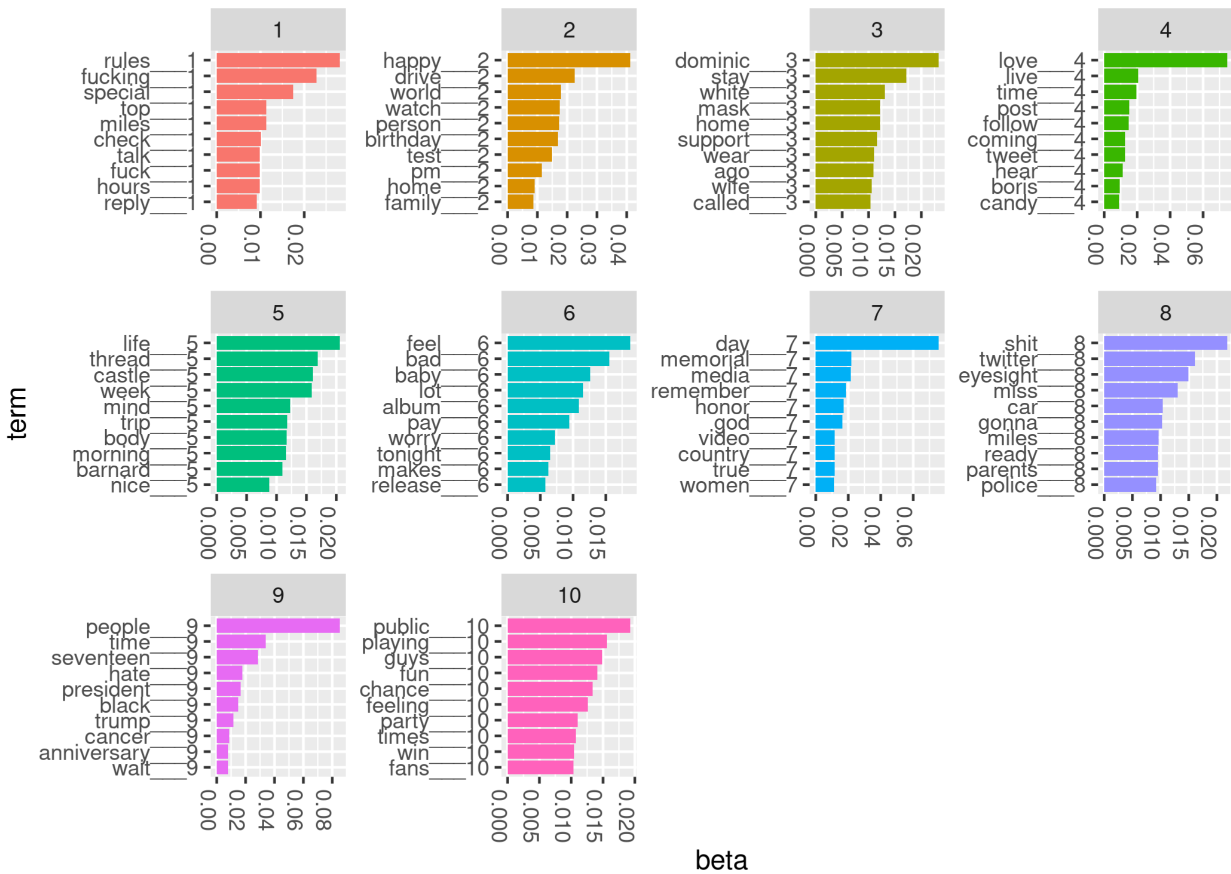

Figure 2: Top terms for LDA model from May 25, 2020 (Memorial Day)

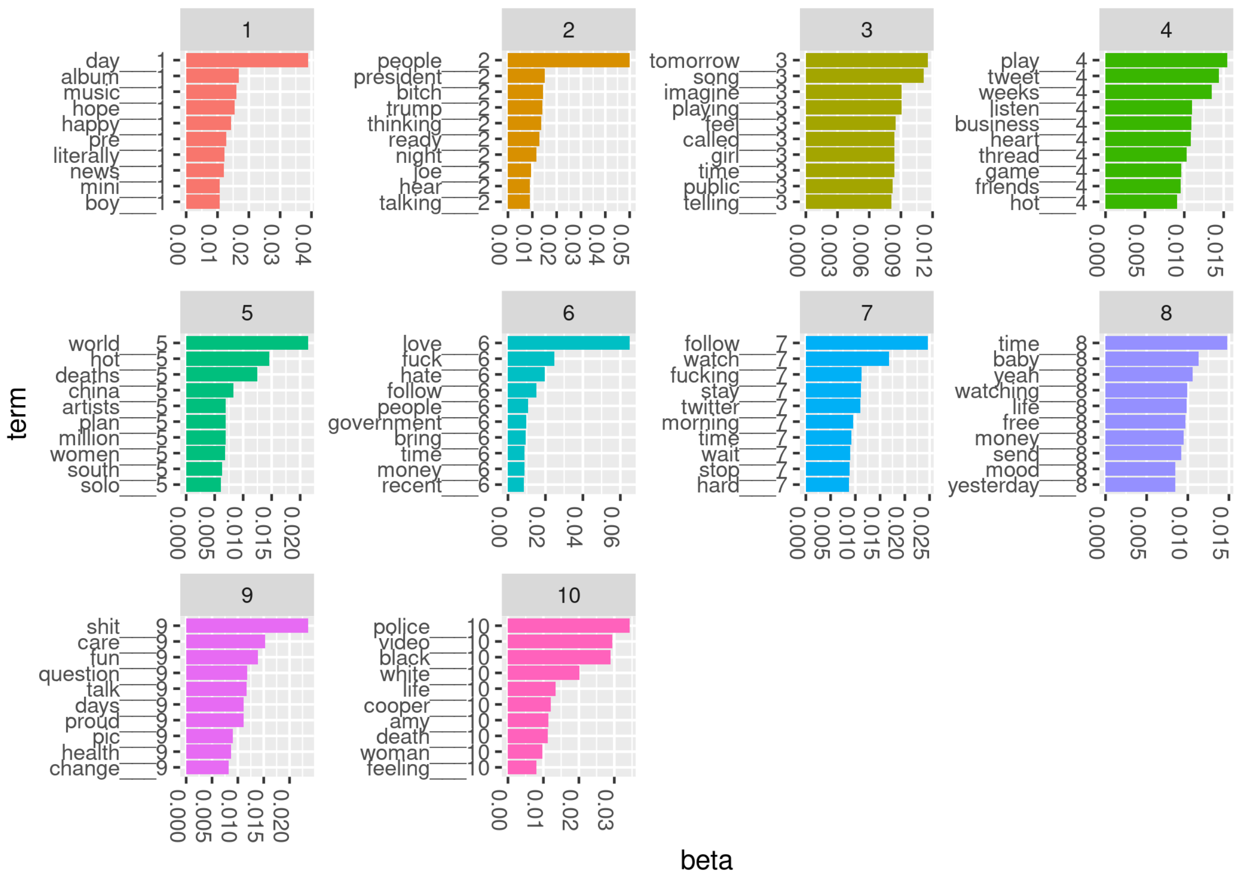

Figure 3: Top terms for LDA model from May 26, 2020

12:00 pm to 12:05 pm (Eastern USA time zone) on May 26, occurred after Floyd's murder, yet we didn't detect this event in our ten-topic LDA model. Several considerations may account for this. While outrage at the murder eventually spread worldwide, there may have been few Floyd-related tweets during our collection time on May 26, less than 24 hours after the murder and video release. Had we extended our analysis to May 27 and beyond, we may have identified George Floyd-related topics.

6 Assessment of learning, exploring more advanced topics, and concluding remarks

6.1 Assessment of learning

We examined student learning with both formative and summative assessments. We conducted formative assessments through weekly discussions with students. In these discussions, we developed action items to advance research progress and overcome challenges. We summatively assessed student achievement at the end of the academic year. Both students wrote a thesis and presented a poster to our statistics department. We asked questions at the poster session to probe student understanding and critically evaluated the theses.

With future students, we will use a written rubric to evaluate theses (Table 2). We'll share the rubric with our students at the start of the academic year.

6.2 Exploring more advanced topics

Twitter data over time inspires a variety of research projects. Supplementing tweets with public data from other sources multiplies the possibilities. For example, one of our two students supplemented tweets with daily stock market index prices. She studied sentiment of finance-related tweets and daily stock market index closing prices (Supplementary files).

Latent Dirichlet allocation modeling and related methods are a major research area in the quantitative social sciences. Advanced students with interest in statistical computing might compare inferential methods for topic models. Those with interests in event detection and time series analysis could build on the findings of our student by explicitly accounting for topic evolution with dynamic topic models (Blei and Lafferty 2006).

6.3 Concluding remarks

Our mentoring in data science aligns with others' calls to reconsider the role of computing in statistics and data science (Carver et al. 2016; Nolan and Temple Lang 2010). Hicks and Irizarry (2018) argue for

Table 2: Rubric for summative assessment of learning objectives.

Learning objective	Assessment item	2 points	1 point	0 points
Write R code to perform text analysis of large volumes of tweets.	R code performs intended analyses	Code contains few or no bugs	Code contains one or more errors	Code contains many errors
Write R code to perform text analysis of large volumes of tweets.	Uses literate programming tools, such as Sweave or knitr	Report is written using literate programming tools. It compiles easily when run by instructor. Time-consuming calculations are cached.	Report is written using literate programming tools, but compilation takes too long or fails.	Report is not written with literate programming tools.
Write R code to perform text analysis of large volumes of tweets.	Uses git for version control	Log reveals regular commits with informative commit messages	Log reveals intermittent commits and uninformative commit messages	Doesn't use git.
Write R code to perform text analysis of large volumes of tweets.	Shares code and data via Github	Instructor easily clones repository from Github. Contains share-able data and instructions for getting other data to reproduce analysis.	One or more needed files is missing from repository.	Doesn't use Github.
Communicate results in a written report and poster presentation.	Organizes poster to highlight main points	When prompted, can describe main points in less than one minute.	Less fluid presentation with periods of silence or confusion.	Disorganized presentation.
Communicate results in a written report and poster presentation.	Accurately presents study and findings during poster session	Fluently describes background, study goals, study design, approach, data, findings, and conclusions	At least one section is incomplete or is verbal explanation is incomplete.	At least one section is missing.
Communicate results in a written report and poster presentation.	Report structure mirrors a research manuscript	Contains abstract, introduction, methods, results, and discussion	At least one section is incomplete.	At least one section is missing.
Translate statistical findings into scientific conclusions.	Places statistical results in their scientific context	Demonstrates understanding of scientific context and integrates findings into it.	Incomplete scientific understanding or incomplete integration of findings.	Major gaps in scientific understanding or integration of findings.
Translate statistical findings into scientific conclusions.	Accurately portrays study limitations	Accurately describes, in writing and in speaking, limitations of the study	Incomplete or partially inaccurate description of limitations	Doesn't describe limitations.
Translate statistical findings into scientific conclusions.	Demonstrates familiarity with relevant literature	Fluent in both relevant data science literature and scientific literature.	Incomplete knowledge and understanding of relevant literature	Major gaps in knowledge and understanding
Develop data science strategies to address a scientific research question.	Presents an original research question	Presents, in writing and in speaking, a novel research question. Explains why it's novel, too.	Partially lacking in elements of question's background or novelty.	Doesn't present an original question.
Develop data science strategies to address a scientific research question.	Effectively uses data visualizations	Visualizations highlight main points of report.	Incomplete or omitted visualizations.	Doesn't use visualizations.
Develop data science strategies to address a scientific research question.	Presents accurate scientific conclusions	Effectively translates analysis results into their scientific context.	Minor inaccuracy in translation of findings into scientific context.	Major errors in translation of results.

incorporating three concepts into data science training: computing, connecting and creating. They use the terms "connecting" and "creating" to describe the processes of applying quantitative methods to real data and research questions and of formulating research questions, respectively. Our tweet analysis projects offer students opportunities in all three skills sets. Our students first formulated research questions, then collected and analyzed data to address the questions. Throughout the projects, students drew heavily on computing, both to acquire data and to analyze it.

Tweet analysis gives students practical experience in the data science process of formulating a research question, gathering data to address it, summarizing the data, visualizing results, and communicating findings. Tweets over time are a rich, large, authentic data set that offers many opportunities. We provided instructions to enable readers to establish their own tweet collections. We also presented details for one analysis strategy. By considering first student research interests and integrating them with our senior thesis learning objectives, we successfully guided two undergraduate researchers in data science research with tweets.

7 Acknowledgements

The authors thank Betsy Colby Davie and Rick Nordheim for helpful discussions and feedback on preliminary versions of the manuscript. We thank the special issue editors and anonymous reviewers for their constructive comments and suggestions.

8 References

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017), "Variational inference: A review for statisticians," *Journal of the American statistical Association*, Taylor & Francis, 112, 859–877.

Blei, D. M., and Lafferty, J. D. (2006), "Dynamic topic models," in *Proceedings of the 23rd International Conference on Machine Learning*, pp. 113–120.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003), "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, 3, 993–1022.

Carver, R., Everson, M., Gabrosek, J., Horton, N., Lock, R., Mocko, M., Rossman, A., Roswell, G. H., Velleman, P., Witmer, J., and others (2016), "Guidelines for assessment and instruction in statistics education (GAISE) college report 2016," AMSTAT.

Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L., and Blei, D. M. (2009), "Reading tea leaves: How humans interpret topic models," in *Advances in Neural Information Processing Systems*, pp. 288–296.

Hicks, S. C., and Irizarry, R. A. (2018), “A guide to teaching data science,” *The American Statistician*, Taylor & Francis, 72, 382–391.

“Introducing JSON” (2020), https://www.json.org/json-en.html.

“Introduction to Tweet JSON” (2020), https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/intro-to-tweet-json.

Kearney, M. W. (2019), “rtweet: Collecting and analyzing Twitter data,” *Journal of Open Source Software*, 4, 1829. https://doi.org/10.21105/joss.01829.

Lin, C. X., Mei, Q., Han, J., Jiang, Y., and Danilevsky, M. (2011), “The joint inference of topic diffusion and evolution in social communities,” in *2011 IEEE 11th International Conference on Data Mining*, IEEE, pp. 378–387.

Nolan, D., and Temple Lang, D. (2010), “Computing in the statistics curricula,” *The American Statistician*, Taylor & Francis, 64, 97–107.

Pelled, A., Lukito, J., Boehm, F., Yang, J., and Shah, D. (2018), “‘Little Marco,’‘Lyin’Ted,’‘Crooked Hillary,’ and the ‘Biased’ media: How Trump used Twitter to attack and organize,” in *Digital Discussions*, Routledge, pp. 176–196.

Porteous, I., Newman, D., Ihler, A., Asuncion, A., Smyth, P., and Welling, M. (2008), “Fast Collapsed Gibbs Sampling for Latent Dirichlet Allocation,” in *Proceedings of the 14th acm sigkdd international conference on knowledge discovery and data mining*, pp. 569–577.

R Core Team (2019), *R: A language and environment for statistical computing*, Vienna, Austria: R Foundation for Statistical Computing.

Robinson, D. (2016), “Text analysis of Trump’s tweets confirms he writes only the (angrier) Android half,” http://varianceexplained.org/r/trump-tweets/.

“Sampled stream” (2019), https://developer.twitter.com/en/docs/labs/sampled-stream/overview.

Silge, J., and Robinson, D. (2016), “tidytext: Text mining and analysis using tidy data principles in R,” *JOSS*, The Open Journal, 1. https://doi.org/10.21105/joss.00037.

Wells, C., Shah, D. V., Pevehouse, J. C., Yang, J., Pelled, A., Boehm, F., Lukito, J., Ghosh, S., and Schmidt, J. L. (2016), “How Trump drove coverage to the nomination: Hybrid media campaigning,” *Political Communication*, Taylor & Francis, 33, 669–676.

Wickham, H. (2019), *Advanced r*, CRC press.

Wickham, H., and Grolemund, G. (2016), *R for Data Science: import, tidy, transform, visualize, and model data*, O'Reilly Media, Inc.

Wiggins, G., and McTighe, J. (2005), *Understanding by Design*.

Wijffels, J., and Belmans, O. (2018), *taskscheduleR: Schedule R Scripts and Processes with the Windows Task Scheduler*.

9 Supplementary files

9.1 Tweets data dictionary

1. Data dictionary

9.2 R code to reproduce the case study

1. tweets.Rmd
2. tweets-one.Rmd
3. recover_tweets.R

9.3 Student projects

1. Student 1 poster: Project_Poster.pdf
2. Student 1 report: report.pdf
3. Student 2 useR 2016 slides: user2016boehm.pdf
4. Student 2 poster: warfdiscovery2016boehm.tiff

9.4 Github repository

1. https://github.com/fboehm/jse-2019