Received: 25 August 2021 | Accepted: 3 January 2022

DOI: 10.1002/sta4.452

SPECIAL ISSUE PAPER

Content and computing outline of two undergraduate Bayesian courses: Tools, examples, and recommendations

Jingchen Hu[1] | Mine Dogucu[2]

[1]Mathematics and Statistics Department, Vassar College, Poughkeepsie, New York, USA

[2]Department of Statistics, University of California Irvine, Irvine, California, USA

Correspondence
Jingchen Hu, Mathematics and Statistics Department, Vassar College, Poughkeepsie, NY, USA.
Email: jihu@vassar.edu

Present address
Jingchen Hu, Box 27, 124 Raymond Ave, Poughkeepsie, NY, 12604

Undergraduate Bayesian education is an area that has started getting attention lately. As many educational innovations and articles are published and increasingly more teaching and learning materials are shared, statistics educators might be interested in incorporating Bayesian statistics in their undergraduate statistics and data science curriculum. In this paper, we share a succinct overview of two undergraduate Bayesian courses we have been teaching, with a comparison analysis to present the similarities and differences in our approaches. We dive deeper into various choices of Markov chain Monte Carlo estimation methods of Bayesian models with a working example and discuss their pros and cons for different learning objectives of computing that aspiring Bayesian educators might have in mind. Furthermore, we share challenges and opportunities for course development and curriculum design. The paper is suitable for aspiring Bayesian educators who are interested in learning ways to introduce Bayesian statistics to undergraduate students.

KEYWORDS
Bayesian computing, Bayesian education, curriculum design, MCMC, statistical computing

1 | INTRODUCTION

Despite the advances in Bayesian computing and the increase in popularity of Bayesian statistics for applied problems, the undergraduate statistics and data science curricula have not yet been able to keep up with these developments. In the top ranked 100 universities and 50 liberal arts colleges, the majority of undergraduate programmes do not offer a Bayesian course; among those that do offer, very few require the course as part of the major (Dogucu & Hu, in preparation).

Even though undergraduate Bayesian education is not where we would like it to be, in the last few years, there have been some important contributions. Several authors have proposed innovative teaching tools at various levels for cultivating Bayesian thinking (Barcena et al., 2019; Eadie et al., 2019; Rouder & Morey, 2019) and incorporating Bayesian computing (Albert & Hu, 2020). For course and curriculum design, some have described their course features and experiences (Johnson et al. 2020), and others have shared their curricula as examples (e.g., Hoegh, 2020; Hu, 2020 ; Witmer, 2017). There are also textbooks published specifically for an undergraduate audience, including Albert and Hu (2019), Reich and Ghosh (2019), and Johnson et al. (2022). Given the heightened interests in introducing Bayesian statistics into undergraduate statistics and data science curricula, we expect the resource pool to keep growing for adding a Bayesian module in existing courses as well as creating full undergraduate Bayesian courses. All these developments also call for existing and aspiring Bayesian educators to further the research of Bayesian education at the undergraduate level.

For aspiring Bayesian educators who want to design a first undergraduate Bayesian course at their institutions, examples and experiences of teaching a semester-long or quarter-long Bayesian course are useful resources (e.g., Hoegh, 2020; Hu, 2020; Johnson et al., 2020; Witmer, 2017). This paper further enriches the resource pool, where we share our curricular approaches to teaching Bayesian statistics to undergraduate students.

TABLE 1 Content and the number of weeks for the 13-week course at Vassar College

Content	Number of weeks
Introduction	1
Bayesian inference for a proportion	1.5
Bayesian inference for a mean	1.5
Gibbs sampler and MCMC	2.5
Bayesian hierarchical modelling	1.5
Bayesian linear regression	2
Exams, additional topics and project presentations	3

We have designed, developed and taught undergraduate-level Bayesian courses at our institutions respectively for the past few years. Both courses are housed in the statistics departments[2]. The unique contributions of this paper include a succinct overview of the two courses, with a comparison analysis to show the similarities and differences in our approaches, including topics and Bayesian computing. We dive deeper into various choices of Markov chain Monte Carlo (MCMC) estimation methods of Bayesian models with a working example and discuss their pros and cons for different learning objectives of computing that aspiring Bayesian educators might have in mind. Furthermore, we share challenges and opportunities for course development and curriculum design. We hope the reader can walk away with increased confidence of designing an undergraduate Bayesian course, as well as tools and approaches that can be used and experimented.

The remainder of the paper is organized as follows. Section 2 provides a succinct overview of the two courses, followed by a comparison analysis in Section 3. We describe and discuss challenges and opportunities for course development and curriculum design in Section 4, and the paper ends with concluding remarks in Section 5.

2 | THE TWO COURSES

2.1 | A 13-week course at a small private liberal arts college

The first course is an upper-level course taught at Vassar College, a private liberal arts college in New York State. The course is a popular elective among students major in mathematics and statistics, especially those in the statistics pathway[3]. It also attracts students from computer science, economics and cognitive science, among other majors. The prerequisites include multivariable calculus and calculus-based probability. Due to the limited statistics offering at the college, no prior statistics exposure is required. No prior statistical computing experience is required either, and additional self-study material on statistical computing in R is introduced to supplement.

The class meets for 150 minutes per week for 13 weeks. Some lectures are used as computing labs. The topics and the number of weeks spent on each topic are shown in Table 1. The course content and structure follow the second half of the *Probability and Bayesian Modeling* textbook by Albert and Hu (2019). In the first part of the course, students learn the basics of Bayesian inference through conjugate models. Moreover, students compare analytical solutions and simulation-based solutions for posterior inference of these conjugate models, where they practice and enhance their statistical computing skills. With these foundations, in the second part, students learn additional multi-parameter models where MCMC algorithms such as Gibbs sampler and metropolis algorithm are necessary for model estimation and posterior inference. In addition, just another Gibbs sampler (JAGS) is introduced (Plummer, 2003), where students learn and practice writing JAGS scripts and discuss the important topic of MCMC diagnostics. In the last part of the course, students learn more advanced Bayesian models, such as hierarchical (multilevel) modelling and regressions. The remaining weeks are spent on exams, additional topics such as latent class modes and course project presentations.

Details of the pedagogical approaches in this course are presented in Hu (2020) for further reading. Teaching and learning material associated with this course, including lecture slides, labs, homework and case studies, can be found online (https://github.com/monika76five/Undergrad-Bayesian-Course).

2.2 | A 10-week course at a large public university

The second course is an upper-level course taught at University of California Irvine, a large public university. The course is a major requirement for a BS in data science and is an elective for statistics minor.[4] In principle, the course has a prerequisite of a year-long calculus-based probability and statistics sequence. In practice, the students enrolled in the course are data science majors who have taken more statistics courses as part of their major, including a year-long sequence of statistical methods for data analysis and a statistical computing course taught in R.

[2] In the case of Vassar College, it is the Mathematics and Statistics Department.

[3] There is currently no major in Statistics at Vassar College. The Mathematics and Statistics major has three pathways: core mathematics, applied mathematics, and statistics.

[4] There is currently no major in Statistics at University of California Irvine.

TABLE 2 Content and the number of weeks for the 10-week course at UC Irvine

Content	Number of weeks
Introduction	1
Balance, sequentiality and conjugacy	3
MCMC	2
Posterior inference and prediction	1
Bayesian linear and logistic regression	2
Exams, hands-on project work, and presentations	1

This course is taught on a quarter calendar with 10 weeks of classes closely following the outline of the *Bayes Rules!* textbook by Johnson et al. (2022). An example schedule is provided in Table 2. Since the students come into the course having covered probability and likelihood, the first part of the course focuses on Bayesian foundations including understanding of the posterior as the balance of the prior and the likelihood. The first part covers the conjugate families beta–binomial, gamma–Poisson and normal–normal. In the second part, the focus is on the posterior simulation analysis including understanding how MCMC works, hypothesis testing and credible intervals. Stan is the main posterior estimation software for the course (Carpenter et al., 2017). In the last part of the course, the focus is mostly on linear models covering multiple linear regression and logistic regression. Further details on the course schedule as well as the slides of the course can be found online (https://www.stats115.com/).

3 | A COMPARISON ANALYSIS OF THE TWO COURSES

As discussed in Section 2 and shown in Tables 1 and 2, the topics covered in the two courses are similar for the most part: both courses spend 1 week on introduction and 3 weeks on foundation of Bayesian inference. From here, they start to diverge a little bit. With 3 more weeks in the semester, the Vassar course invests slightly more time on MCMC. It also gets to introduce Bayesian hierarchical modelling before regressions. With only 10 weeks, the UCI course chooses to introduce logistic regression instead of hierarchical modelling, after covering linear regressions. In general, both courses follow the three topic areas identified by Dogucu and Hu (in preparation): (1) foundation of Bayesian inference, (2) Bayesian computing, and (3) Bayesian modelling, based on analysis of 29 syllabi of undergraduate Bayesian courses.

The bigger difference between the two courses lies in their computing approaches, which we now turn to.

3.1 | Comparison of Bayesian computing approaches

Choosing appropriate computing tools for a Bayesian course is an important and delicate task. Albert and Hu (2020) provide an extensive review of Bayesian computing for an undergraduate audience: non-simulation approaches, Monte Carlo simulation approaches, and MCMC algorithms. Here, we focus on discussing our computing choices for MCMC in the two undergraduate Bayesian courses.

Both courses have a strong focus on simulation-based teaching and learning. Before introducing MCMC algorithms for advanced multi-parameter Bayesian models (e.g., regressions and hierarchical models), we teach Monte Carlo simulation approaches for conjugate models with R, where analytical solutions are available. This provides an ideal learning environment for students to grasp simulation-based approaches, which can be compared to the available analytical solutions. Students are exposed to programming techniques such as simulating a number of random draws from a known posterior distribution and working with posterior summaries of simulated parameter draws. Next, students get to write MCMC algorithms for relatively simple multi-parameter Bayesian models, including the Gibbs sampler and/or the Metropolis and/or the Metropolis–Hastings algorithm. This is where students practice writing loops for iterative sampling, reject or accept using *if* statements, perform MCMC diagnostics, among other things.

Moving to advanced multi-parameter models requires more and varied MCMC algorithms, and for this purpose, the two courses have made different MCMC software choices.

The Vassar course uses `runjags` (Denwood, 2016), an R package for the JAGS software for MCMC estimation. Other R packages include `coda` and `ProbBayes` (Albert, 2020). Once students install JAGS on their computers, installing the `runjags` package is straightforward. We choose JAGS since its syntax is descriptive of the Bayesian model: In a standard JAGS model script, the user provides a sampling section for the model and a prior section which includes the prior distribution for each model parameter. Its descriptive nature connects learning JAGS to students' understanding of the model itself, while writing JAGS scripts reflects and reinforces students' learning and understanding of the model. Posterior inference, including posterior summaries and posterior prediction, requires students to write appropriate R scripts by themselves, which further practice their programming skills. In fact, in this course, JAGS and `runjags` are all we cover as MCMC estimation tools. With ample sample JAGS scripts and a clear introduction of the model, our experience has shown that students are able to write JAGS scripts for more advanced Bayesian modelling techniques such as latent class models, and perform appropriate posterior inference by writing additional R scripts. The course has a strong focus on and a learning objective of expressing Bayesian models through JAGS and performing subsequent posterior analysis; therefore, we do not introduce additional packages that can perform MCMC estimation more directly, tools that are introduced in the second course.

The UCI course uses rstan (Stan Development Team, 2020), an R package for the Stan software, as well as other R packages including rstanarm (Goodrich et al., 2020), bayesrules (Dogucu et al., 2021), and bayesplot. Similar to JAGS, the Stan syntax requires a detailed script defining the model. For each type of model, students write Stan syntax for posterior simulation. Stan syntax consists of three blocks: data, parameters, and model (likelihood and the prior). Once students get a good grasp of different components of the model and the Stan syntax, we move onto using the rstanarm for further practice with posterior simulation. The rstanarm package has a simpler syntax with resemblance to other R modelling frameworks. Moreover, supplementary packages such as bayesplot play well when it comes to posterior analysis and prediction. In short, the course adopts a philosophy that students should know what is going on behind functions of R packages. They should be able to compute a statistic (e.g., posterior predictive credible intervals) without relying on built-in packages. Once they grasp the conceptual understanding of the topic, then they start relying on the built-in packages in the R ecosystem.

3.2 | Analysis of MCMC estimation approaches

Section 3.1 describes the different MCMC estimation choices we have made for the two courses and discusses our rationale when making these choices. In this section, we provide illustration of the three aforementioned MCMC estimation choices with a working example: a Bayesian simple linear regression of the price of a house sale given its area size, taken from Chapter 11 of *Probability and Bayesian Modeling* (Albert & Hu, 2019).

Suppose that PriceArea contains the dataset, and the column names for these two variables are price and area, respectively. We first present the JAGS and Stan approaches, side-by-side, to better show their similarities and differences. Both approaches have two components: (1) define the model, and (2) simulate the posterior. The first component describes the Bayesian model, where JAGS code and Stan code are used respectively. The second component is for posterior simulation, realized through an R package, runjags and rstan, respectively.

```
### JAGS

# STEP 1: DEFINE the model
modelString <-"
model {
## sampling
for (i in 1:n){
   Y[i] ~ dnorm(beta0 + beta1*X[i], invsigma2)
}
## priors
beta0 ~ dnorm(0, 0.0001)
beta1 ~ dnorm(0, 0.0001)
sigma ~ dexp(0.0008)
invsigma2 <- 1 / (sigma * sigma)
}"

# STEP 2: SIMULATE the posterior
posterior <- run.jags(model = modelString,
                      data = list(Y = PriceArea$price,
                                  X = PriceArea$area,
                                  n = nrow(PriceArea)),
                      monitor = c("beta0",
                                  "beta1",
                                  "sigma"),
                      adapt = 1000,
                      burnin = 5000,
                      sample = 5000,
                      thin = 1,
                      n.chains = 1,
                      inits = initsfunction)
```

```
### Stan

# STEP 1: DEFINE the model
model <- "
  data {
    int<lower = 0> n;
    vector[n] Y;
    vector[n] X;
  }
  parameters {
    real beta0;
    real beta1;
    real<lower = 0> sigma;
  }
  model {
    Y ~ normal(beta0 + beta1 * X, sigma);
    beta0 ~ normal(0, 100);
    beta1 ~ normal(0, 100);
    sigma ~ exponential(0.0008);
  }
"

# STEP 2: SIMULATE the posterior
posterior <-  stan(model_code = model,
                   data = list(Y = PriceArea$price,
                               X = PriceArea$area,
                               n = nrow(PriceArea)),
                   warmup = 5000,
                   iter = 10000,
                   thin = 1,
                   chains = 1,
                   seed = 84735)
```

In the first component for defining the model, Stan has one more block of data compared to JAGS. Moreover, JAGS lists the prior distributions for each parameter line by line, separated from the sampling model, whereas Stan defines the parameters first in the parameter block and the prior choices are specified in the model block, together with the sampling model. There are also minor differences in how distributions are specified; for example, JAGS uses mean and precision for a normal distribution, while Stan uses mean and standard deviation.

The second components for simulating posterior with `runjags` and `rstan` R packages are similar for the most part. Both require the model object from the first component, the data list and the number of MCMC chains to be run, although certain argument names may differ. `runjags` allows the specifications of `adapt`, `burnin`, and `thin` of each MCMC chain, and its `sample` refers to the number of posterior samples post burnin and thinning. `rstan`, on the other hand, uses `warmup` instead of burnin and includes the warmup period and the sample total in the `iter` argument. One can choose to list `warmup` and `iter` separately as in our sample script above. It also allows setting of `thin`, although `thin = 1` is the default and the recommended value due to the efficiency of the sampler, and therefore, it is rarely tuned (Stan Development Team, 2020). However, setting a seed is much more straightforward in `rstan` than in `runjags`: the former requires a provided seed, whereas the latter needs a separate chunk (excluded for brevity) that specifies the seed through the `inits` argument. Moreover, `runjags` requires users to include the parameters to be tracked and saved in the `monitor` argument; if a parameter is not included, its posterior draws will not be available in the output.

Overall, the JAGS and Stan approaches are similar for our working example of estimating a Bayesian simple linear regression model. They share many similarities in other commonly used Bayesian models as well.

Next, we provide the syntax for the same model using `rstanarm`, an R package containing wrapper functions for implementing Stan. The `rstanarm` syntax mimics R modelling functions closely (e.g., `glm()`); however it specifies a Bayesian model and uses Stan for posterior estimation. The syntax of `rstanarm` includes every component of the posterior simulation as shown earlier for JAGS and Stan. Since the `rstanarm` syntax mimics other R modelling functions, it is easier to read and follow for R users, especially those with prior experience with using R to fit regression models. In a way, `rstanarm` translates from R syntax to Stan syntax on the backend. This creates advantages for learning, including a simpler syntax that is easier to write and debug. In addition, `rstanarm` produces a simpler object in the output that also makes the posterior analysis easier from a computing perspective. The modelling toolkit of `rstanarm` is simple to use and comprehensive for undergraduate students in a first Bayesian course; however, we often expect our students to be able to read other resources and also be able to fit more complex models in their final projects and in their future careers. Given these goals, being solely dependent on wrapper functions that `rstanarm` provides can be limiting. Therefore, the UCI course teaches the Stan approach to pave the way for more complex modelling but also utilizes `rstanarm` when the focus is on not on the posterior simulation itself but on analysing the posterior. Overall, the use of `rstanarm` saves time in learning. Moreover, when students are exposed to both `rstanarm` and Stan, they have the opportunity to learn more complex models.

```
### rstanarm

posterior <- stan_glm(price ~ area,
                      data = PriceArea,
                      family = gaussian,
                      prior_intercept = normal(0, 100),
                      prior = normal(0, 100),
                      prior_aux = exponential(0.0008),
                      chains = 1,
                      warmup = 5000,
                      iter = 10000,
                      seed = 84735)
```

4 | CHALLENGES AND OPPORTUNITIES

For aspiring instructors who are interested in designing a Bayesian course of their own, we share a few challenges and opportunities in this section. One part of the sharing is borrowed from the broader statistics education literature not specific to Bayesian courses, such as using real-world data (GAISE, 2016) and bringing modern computing into the statistics curricula (Nolan & Lang, 2010). The other part of what we share is based on our own experiences of teaching and continuous interaction with undergraduate students who are learning Bayesian statistics.

4.1 | Real-world applications

Despite its focus on the introductory statistics courses, *the Guidelines for Assessment and Instruction in Statistics Education (GAISE) College Report* (GAISE, 2016) provides recommendations that can extend beyond the introductory course. Recommendation on integration of real data with a context and purpose from this report finds its place in our Bayesian courses. In both courses, we introduce a variety of applied problems using real data for different course activities.

First and foremost, our lecture and lab examples come from real-world datasets. These datasets are available in the R packages accompanying the course textbooks; Albert (2020) and Dogucu et al. (2021). The topics of the datasets are from a variety of fields, including deaths after heart attack at different hospitals, body features of penguins, weather data (e.g., snowfall amount and whether rains or not), and US women labour participation. Second, real-world datasets appear in homework, exams and other data analysis activities. In addition, students get to work on projects, which usually involve real datasets of their own choices. They receive guidance on where and how to find datasets as well as a confirmation that they can work on the dataset that they picked.

4.2 | Bayesian computing

There are pros and cons of each reviewed MCMC estimation approach in Section 3. Readers should choose their preferred MCMC approach(es) given their learning objectives and student background, among other things.

As discussed, JAGS and Stan and their corresponding R packages are overall similar in syntax and uses. Therefore, their teaching approaches are also similar. Both are open-source and have online forums for discussion.[5] There exist Python interfaces for both as well (called `PyJAGS` and `PyStan`) if an instructor wishes to use Python instead of R.

As Section 3.2 illustrates, one benefit of Stan is that there are wrapper packages such as the aforementioned `rstanarm`, as well as `brms` (Bürkner, 2017), whereas there are no wrapper functions of JAGS to the best of our knowledge. Therefore, an instructor utilizing Stan has the opportunity to decide whether to use a wrapper package and, if so, whether the wrapper packages would be taught prior to Stan or after Stan. The UCI course first teaches Stan then `rstanarm`. The rationale behind is to first detail the posterior simulation process with Stan earlier in the course and then to use the posterior simulations with `rstanarm` for posterior analysis later. However, the reversed order is possible and should be considered especially when teaching students who use R but with limited computing preparation beyond R. In this way, students can first get comfortable with simple models and a familiar syntax then move to more complex models with a new syntax. It is important to note that in this paper, we focus on full Bayesian courses, and mainly for statistics and data science undergraduate majors. Instructors who teach Bayesian methods as a module within existing courses, or in a field with a strong applied focus (e.g., Bayesian econometrics) may also consider using solely the wrapper packages if teaching new syntax is not a priority.

Another distinction between JAGS and Stan which might be relevant to some instructors is that JAGS implements commonly used MCMC algorithms (such as Gibbs, Metropolis and Metropolis–Hastings). Stan, on the other hand, implements the Hamiltonian Monte Carlo algorithm (Betancourt, 2017), which could be an additional topic in a Bayesian course where Stan is the main MCMC estimation approach.

A challenge we have faced in teaching Bayesian statistics with `rstan` is a variety of installation problems that students face. We especially encountered this problem in Spring 2020 with the most recent major update of R to 4.0.0 and have not encountered it in our recent academic terms. For such foreseeable installation issues, if possible, we recommend that instructors get support from IT professionals with either helping students debug installation problems or providing R with `rstan` on a local server. We have seen both of the methods to work with benefits and challenges of each. The latter is a better solution for larger classes in our experiences.

4.3 | Student-centred learning

We believe in student-centred learning. It could pose challenges, especially for our Bayesian courses where students rarely have any prior exposure to the topic and thus there always seems to be ‘too much’ new material to cover in a course. Nevertheless, we have found a few opportunities in fostering student-centred learning in undergraduate Bayesian courses.

Course projects are becoming popular in undergraduate statistics and data science courses. In both Bayesian courses, we regard student projects as important components (25% and 15% of the final grade in the two courses, respectively). We share the philosophy of making these student projects completely student-driven: either individually or in a small group, student identify research questions of their own interest, find appropriate data sources, develop suitable Bayesian modelling techniques, and perform various posterior inferences that can help answer their research questions. Guidance and help are provided throughout the entire process, as well as checkpoints (e.g., project proposal and project draft).

[5] The JAGS forum: https://sourceforge.net/p/mcmc-jags/discussion/; the Stan forum: https://mc-stan.org/community/.

Students are encouraged to start thinking about their projects as early as first day of classes (e.g., showing sample projects from previous students).

Student projects are not the only student-centred learning opportunities. Bayesian modelling topics stimulate a wide range of applications, some of which can be designed as open-ended case studies to encourage students to freely explore: modelling strategies, programming techniques, and posterior inference approaches, among other things. Moreover, when properly selected, accessible journal articles can be introduced in Bayesian courses, where students read, discuss and critique scholarly work. Curricular designs such as creating reading guides and computing labs based on simulation studies in these journal articles are what we have found highly effective approaches to enriching students' learning experience. Interested readers are referred to sect. 3 of Hu (2020) for more details on introducing case studies and journal articles with examples.

4.4 | Conceptual understanding

Over the years, we have identified a few shared conceptual and practical difficulties undergraduate students are usually faced with when learning Bayes. First and foremost is understanding priors and then subsequently how to choose priors. Second, understanding and visualizing multivariate probability distributions are no easy task, and a lack of it might impede students developing their skills to perform posterior derivation of multi-parameter Bayesian models (e.g., when coding their own MCMC algorithms).

Also, students might come to a Bayesian course with diverse backgrounds. While the UCI course is mostly taken by data science majors, the Vassar course has seen students of all kinds of majors: other than the common majors of mathematics and statistics, computer science, economics and cognitive science, students major in political science, environmental studies, and occasionally drama, philosophy and history have chosen to take the Bayesian course. Catering for such a diverse group can be challenging but also fulfilling, especially seeing a pool of student projects with many distinctive topics and focuses.

A prior understanding that students often need in our Bayesian courses is multivariable calculus, especially multiple integrals. For many students, this is an opportunity as they get to use their calculus knowledge in a real data context. However, for few who have a weaker understanding of these topics, this can also be a challenge. In our courses, we utilize calculus for derivations but we also iterate understanding how calculus is being used rather than focusing on the detailed calculus formulae. Depending on the calculus weight in a given course and the background of students, instructors may possibly need to provide calculus review opportunities.

4.5 | Comparisons to frequentist methods

Another important and delicate decision to make is whether comparisons between frequentist and Bayesian methods should be introduced and discussed, and if so, how. We think the decision depends highly on the course itself, and we share how we have made our decisions for our Bayesian courses.

The Vassar course does not require students' prior exposure to any statistics course, which leads to the decision of no comparison between frequentist and Bayesian methods in the course. However, students with prior exposure to frequentist methods are always very curious about the comparison, which shows up in classroom interaction and more notably, in their course projects. In the UCI course, students come to class having taken multiple statistics courses with the frequentist framework. Therefore, comparisons of Bayesian and frequentist methods are covered, starting on Day 1 with an ungraded quiz assessing whether students have frequentist or Bayesian approaches. Later on, we discuss comparisons of the two paradigms for hypothesis testing and confidence versus credible intervals.

For a more in-depth reference on such comparisons, Wakefield (2013) provides examples and discussions for various regression models.

5 | CONCLUDING REMARKS

With an increasing number of computational and pedagogical tools available, we believe that undergraduate Bayesian courses will become more common. We hope the content and computing outline of our two Bayesian statistics courses, together with our shared tools, examples and recommendations, would help instructors brainstorm designing a full Bayesian course for their undergraduate students.

In closing, we invite inspiring Bayesian educators to join a friendly online community for undergraduate Bayesian education online (https://undergrad-bayes.netlify.app/). The associated Slack workspace fosters discussions of teaching Bayesian statistics to undergraduates. The site also includes lists of resources of textbooks, courses and papers, among other things. We have another resources page online (https://monika76five.github.io/Undergrad-Bayesian-Education-Resources/) with a similar purpose.

ACKNOWLEDGEMENT

Dogucu was supported by NSF grant IIS-2123366.

ORCID

Jingchen Hu https://orcid.org/0000-0002-4283-181X

REFERENCES

Albert, J. (2020). ProbBayes: Probability and Bayesian modeling. Retrieved from https://github.com/bayesball/ProbBayes, R package version 1.1.

Albert, J., & Hu, J. (2019). *Probability and Bayesian modeling*: CRC Press. https://monika76five.github.io/ProbBayes/

Albert, J., & Hu, J. (2020). Bayesian computing in the undergraduate statistics curriculum. *Journal of Statistics Education*, *28*, 236–247.

Barcena, M. J., Garin, M. A., Martin, A., Tusell, F., & Unzueta, A. (2019). A web simulator to assist in the teaching of Bayes' theorem. *Journal of Statistics Education*, *27*, 68–79.

Betancourt, M. (2017). A conceptual introduction to Hamiltonian Monte Carlo. *arXiv preprint*.

Bürkner, P.-C. (2017). brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, *80*(1), 1–28.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., & Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of statistical software*, *76*(1), 1–32.

Denwood, M. J. (2016). runjags: An R package providing interface utilities, model templates, parallel computing methods and additional distributions for MCMC models in JAGS. *Journal of Statistical Software*, *71*(9), 1–25.

Dogucu, M. & Hu, J. The current state of undergraduate Bayesian education and recommendations for the future, arXiv preprint.

Dogucu, M., Johnson, A. & Ott, M. (2021). bayesrules: Datasets and supplemental functions from Bayes rules! book. Retrieved from https://github.com/bayes-rules/bayesrules, R package version 0.0.2.

Eadie, G., Huppenkothen, D., Springford, A., & McCormick, T. (2019). Introducing Bayesian analysis with m&m's: An active-learning exercise for undergraduates. *Journal of Statistics Education*, *27*, 60–67.

GAISE (2016). Guidelines for assessment and instruction in statistics education (GAISE): College report. http://www.amstat.org/education/gaise

Goodrich, B., Gabry, J., Ali, I., & Brilleman, S. (2020). rstanarm: Bayesian applied regression modeling via Stan. Retrieved from https://mc-stan.org/rstanarm, R package version 2.21.1.

Hoegh, A. (2020). Why Bayesian ideas should be introduced in the statistics curricula and how to do so. *Journal of Statistics Education*, *28*, 222–228.

Hu, J. (2020). A Bayesian statistics course for undergraduates: Bayesian thinking, computing, and research. *Journal of Statistics Education*, *28*, 229–235.

Johnson, A. A., Ott, M., & Dogucu, M. (2022). *Bayes rules! An introduction to Bayesian modeling with R*. CRC Press. https://bayesrulesbook.com

Johnson, A. A., Rundel, C., Hu, J., Ross, K., & Rossman, A. (2020). Teaching an undergraduate course in Bayesian statistics: A panel discussion. *Journal of Statistics Education*, *28*, 251–261.

Nolan, D., & Lang, D. T. (2010). Computing in the statistics curricula. *The American Statistician*, *64*(2), 97–107.

Plummer, M. (2003). JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling. In *Proceedings of the 3rd International Workshop on Distributed Statistical Computing*, *124*, Vienna, Austria, pp. 1–10.

Reich, B., & Ghosh, S. (2019). *Bayesian statistical methods*: CRC Press. https://bayessm.wordpress.ncsu.edu/

Rouder, J. N., & Morey, R. D. (2019). Teaching Bayes' theorem: Strength of evidence as predictive accuracy. *The American Statistician*, *73*, 186–190.

Stan Development Team (2020). RStan: the R interface to Stan. Retrieved from http://mc-stan.org/, R package version 2.21.2.

Wakefield, J. (2013). *Bayesian and frequentist regression methods*: Springer.

Witmer, J. (2017). Bayes and MCMC for undergraduates. *The American Statistician*, *71*, 259–264.

How to cite this article: Hu, J., & Dogucu, M. (2022). Content and computing outline of two undergraduate Bayesian courses: Tools, examples, and recommendations. *Stat*, *11*(1), e452. https://doi.org/10.1002/sta4.452