# Introductory Data Science: A Blueprint to Navigate Curricular, Pedagogical, and Computational Challenges

Elijah Meyer
Department of Statistics, Duke University
and
Mine Çetinkaya-Rundel
Department of Statistics, Duke University

May 26, 2023

**Abstract**

The text of your abstract. 200 or fewer words.

*Keywords:* Data Science, Curriculum, Pedagogy

# 1 Audience

Audience - new to teaching / teachers increasing course size (mention this as a draw in because this is happening). This could be a hook.

I think this needs to be further brought out in the beginning of the intro when formalling writing….

# 2 Introduction

Reader should want to continue reading by thinking "this is something that's interesting"

– **Why** we are writing this paper

There is a demand for data scientists. An estimated 11.5 million new data science jobs are projected to be created by 2026, while employment of data scientists is projected to grow by 36 percent from 2021 to 2031 (BLS 2022). **As demand increases, class sizes to best prepare students for these positions increases as well**. The increasing volume of enrollment of data science students (Redmond 2022) requires that statistics and data science educators commit to developing modern curriculum in order to help students be successful. Despite the demand, colleges are still struggling with what a modern data science curriculum should look like (Schwab-McCoy et al. 2021), and how it can be effectively taught to a large student audience. To this point, much more thought, work, and discussions need to take place there before a consensus is reached on what should a modern data science curricula should be.

**What people have been investigating / what's recommended**

Curricular

- College of Charleston, South Carolina (Anderson et al. 2014)
- Intro to Data Science implementation (Asamoah et al. 2015)
- Curriculum Guidelines for Undergraduate Programs in Data Science (De Veaux et al. 2017)
- 50 Years of Data Science (Donoho 2017*a*)
- Association for Computing Machinery (ACM) Education Council (Danyluk et al. 2021)

Pedagogical

- Computational and Tech recommendations in curricula (Donoho 2017*a*, Beckman et al. 2021)
- Active Learning / Other recommendations (Dogucu & Çetinkaya Rundel 2022, Çetinkaya Rundel et al. 2022) + GAISE (Carver, 2016); Conference Board of Mathematical Sciences, 2016).

**But...**

However, the majority of context across the landscape of data science curricula largely focuses on how to model data (Donoho 2017*b*). This, in combination when a lack of consensus on what constitutes a data science curricula, and even less research on how to develop modern data science course presents a need for a blueprint to design and implement and modernized introduction to data science course.

**So**

The need for those who can make meaning of data are clear. Thus, it is critical that data science education provides programs that adequately prepare and train students in the

field of data science. This article adds to the existing literature by describing a modernized introductory data science course and how its taught, at Duke University.

The purpose of this paper is to offer valuable structure, while providing experiences from our perspective. It is of importance that we stress the amount of flexibility and strength in individuality in all aspects when creating, designing, and implementing an introductory data science course. (I don't know where this goes yet; Mine brings up a good point to make sure this is somewhere highlighted at the forefront)

In this paper, we discuss the integration of technology in our Introductory to Data Science course, and how these choices have helped shape our curricular and pedagogical decisions made. This includes detailing the implementation of the Kaplan Way learning model, to support a large class of students with a diverse background in statistics, data science, and coding experience. Within this format, we provide examples of and describe activities and assessments given both in and outside of class. We extend discussions and provide recommendations for implementing and integrating computing tools, such as R-studio and GitHub, through our experiences in our course. Lastly, we discuss challenges, and provide insight to help faculty wanting to adopt or adapt a course similar to introductory to data science at Duke University. The purpose of this paper is to continue the discussion, and present a modernized curricula for an introductory data science course at Duke University, and the pedagogical decisions to help best equip students with the data science skills necessary for future classes.

(March 1st - started writing here)

# 3 The Course

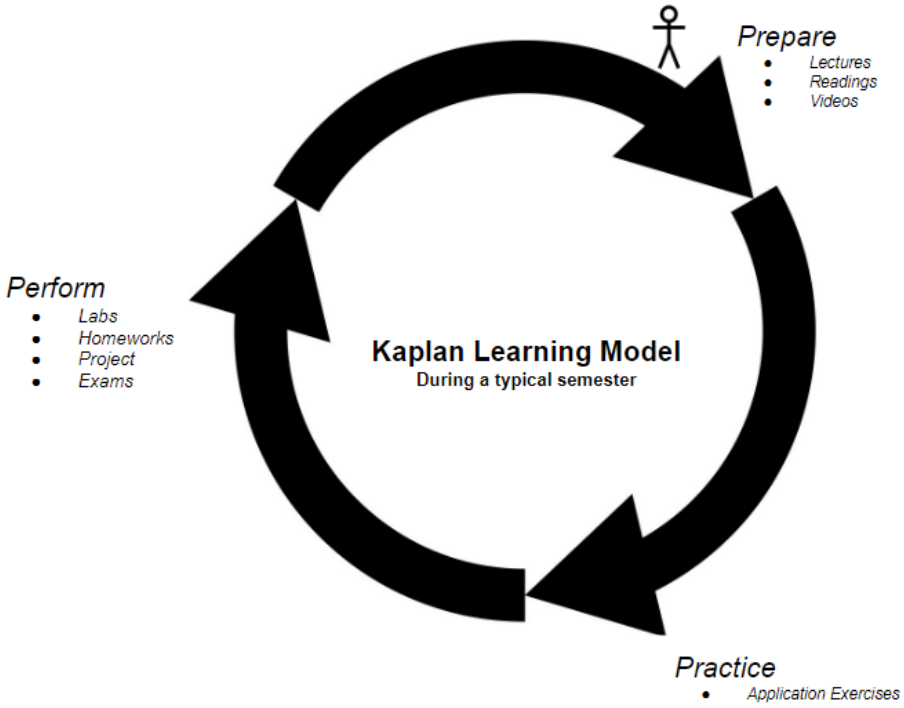
Reader should want to continue reading by thinking "this is something that I would like to teach"

In the following sections, we describe our introductory data science course offered to predominantly freshman level students at Duke University called Introduction to Data Science and Statistical Thinking (STA199). Often, these students are (interested in / major + minor in / description of typical student demographic). Students enrolling in STA199 commonly have little to no statistics, data science, or coding experience. In a typical semester, this course seats roughly 179 students, which is considered to be large by all measures. Both the lack of experience and size of class are identified as two common hurdles by faculty when trying to implement an introductory data science course (Schwab-McCoy et al. 2021, Kokkelenberg et al. 2008). However, by the end of this course, students are able to use data both R and GitHub to understand natural phenomena, investigate patterns, and model outcomes in a reproducible format. \

This course is built on four large-scale learning objectives: learn to explore, visualize, and analyze data in a reproducible and shareable manner; gain experience in data wrangling and munging, exploratory data analysis, predictive modeling, and data visualization; work on problems and case studies inspired by and based on real-world questions and data; learn to effectively communicate results through written assignments and project presentation. These objectives are accomplished through interactive lectures and labs that present content, problems, and case studies inspired by and based on real-world questions and data.

When teaching, instructors are committed to the Kaplan Way learning model "that combines a scientific, evidence-based design philosophy with a straightforward educational

approach to learning" (Schweser 2023, pg. 3).




This model is composed of three phases: prepare, practice, and perform. All three phases are designed to guide the instructor in facilitating an overall quality learning experience for the student. Each of these phases are performed during a typical week within a semester, including through preparation material, lectures, in-class application exercises, labs, and assessments such as homework.

During the prepare phases, students are completing readings, watching videos, and listening to lecture that ultimately builds upon a new foundation of data science concepts being learned. The goal of preparing students is to put them in a position where they can build upon what they are learning, and create new knowledge through the practice phase. The practice phase is designed to be an opportunity for students to reinforce the new information gained, as well as uncover new concepts in data science. This is achieved through the use of interactive application exercises (AEs) in class where students work alone, in groups, or

with the instructor in live coding sessions. Finally, students enter the perform phase to show their progress made in the previous two phases. This is typically done through assessments such as homework, exams, and projects. Additionally, students perform individually, or with a group, during weekly labs that tend to focus more on computation. This learning model is a continuous cycle throughout the semester as new topics are introduced.

Topics taught in STA199 fall under two major units: Unit 1 - Exploring data; Unit 2 - Making rigorous conclusions. In Unit 1, students become first introduced to R, R-studio, and Github. During this unit, students start to create data visualizations and learn how to both import and manipulate data to be better suited for analysis. In unit 2, students extend their investigations with data to include modeling. Specifically, students fit a variety of models (simple linear regression, multiple linear regression, logistic regression), and learn the fundamentals of hypothesis testing. For a more complete description about the topics taught and data sets used in creating these lessons, please see **A fresh look at introductory data science (cite)**.

In this paper, we describe the preparation and implementation process of STA199 in its entirety. This includes details of the teaching team used to instruct STA199, technology chosen to use when creating and facilitating our introductory data science course, and the pedagogical choices that go into a typical week of teaching. This includes a comprehensive description of a flexible framework on how to create, set up, and implement an introduction to data science course similar to STA199. When describing this framework, we articulate first hand experiences and suggestions surrounding some of the choices made to create and instruct STA199.

# 4 Teaching Team

We define a teaching team as a group consisting of one instructor and multiple teaching assistants (TAs). The assignment of any teaching assistant is to both support the instructor in charge of the class, and support the students in the classroom. These TAs range from undergraduate to PH.D. level students, and vary in teaching experience. (Writing on TA selection process). Once selected… (writing on TA training).

Once training is complete, students are assigned roles that indicate their responsibilities during the semester. These roles include *course organizer*, *head TA*, *lab leader*, and *lab helper*. Often, these roles are given based on the level of student, with more academically experienced TAs taking on the roles of course organizer, head TA, and lab leader, where as students with less experience (i.e. undergraduate students) take on the role of lab helper.

Lab sections are held once a week, and are facilitated in person by both a lab leader and lab helper. The responsibilities of a lab helper are supporting both the students and lab leader as they see fit. Examples of this may include setting up the classroom before class, or conducting small group conversations when students have questions about the material. The lab leader is responsible for facilitating the lab. This involves working through a pre-made lab to ensure they can help students apply concepts discussed in lecture to what's being assigned. In addition, both must hold two hours of office hours each week and have grading responsibilities assigned throughout the semester.

Head TA responsibilities can generally be categorized as one of the following: Administrative or Pedagogical. Administrative responsibilities include the organization and distribution of TA responsibilities throughout the semester. It is imperative that the head TA and instructor clearly communicate expectations with each other to establish exactly how rules

and responsibilities that are given to the TAs are assigned. Administrative duties include reminding other TAs about bi-weekly payroll deadlines and ensure TAs are working their alloted hours per week (and not more). Within these alloted hours, a head TA distributes grading assignments and deadlines to both lab helpers and leaders per week. They make sure all TAs complete grading within a week and spot check the grading accuracy and quality of all written feedback given. (insert 1-2 sentences about experience). Pedagogically, head TAs are responsible for creating or reviewing answer keys and grading rubrics for homework and lab assessments as the instructor sees fit. Each head TA is also assigned to instruct one lab section during the semester. Before becoming a Head TA, there are additional

The course organizer is expected to work across each section of STA199, instead of working with a single instructor. Their responsibilities include creating rubrics for and working through homework and lab assignments. Additionally the course organizer, along with the instructor, answers real time questions virtually during labs from lab leaders. Questions often range from content related to technical questions about GitHub and R. Finally, the course organizer is responsible for handling all requested assignment extension requests from students. This includes filing away student exemptions, providing extensions for extreme circumstances, and enforcing the late work policy outlined in the syllabus when necessary.

(Paragraph on flexiability in team structure)

Through my experience as an instructor working with this designed team, it is imperative that everyone is communicating with each other. A team with many different roles poses risk for the instructor to be unaware of how or what decisions are being enacted at the grading and lab levels of the course. Thus, it is recommended that the instructor trains everyone on the teaching team to use a communication system that allows every member

to communicate any questions they may have, or decisions they make to the entire team. In the past, we have used the software *Slack*, with appropriately named channels such as *grading-questions* where TAs can post examples and questions about grading and the instructor can clearly state their expectations. Further, it should be noted that the head TA should not be treated as a "bridge of communication" for the instructor to the rest of the teaching team. It is critical that that the instructor is in consistent contact with all members of the teaching team in making sure all lab leaders and helpers understand the course content, upcoming assignments, and know what's expected of them in their assigned role. We recommend holding a weekly meeting with all members of the teaching team to ensure this. When members are unable to come, it is an expectation that they watch a recorded video of the meeting and reach out if they have any questions about what was discussed.

# 5 Technology

In this section we will detail the computing infrastructure used in STA199 used to create and a course such as STA199. This includes details on how to use R, R-studio, and GitHub's from the instructor's perspective to set up lectures, AEs, labs and homework. First, we start with the student information necessary to collect and steps that must be taken before creation can take place.

## 5.1 Setting up a GitHub account

We can use R, R-studio, and GitHub to create interactive lessons, and assign pre-created assessments to individual or groups of students. To do so, we must instruct students to create a GitHub account. This is done of the first day of class, and often, students are given

time during class to sign up. Following tips from "Happy Git with R" (cite), we suggest students do the following when creating their name:

– Incorporate your actual name – Reuse your username from other contexts if you can – Pick a username you will be comfortable revealing to your future boss – Be as unique as possible in as few characters as possible. Shorter is better than longer – Avoid words with special meaning in programming (i.e., NA)

Once students create their account, we suggest getting this information from them in a survey. This normally can be done through your learning management system. It is critical to reiterate to students that spelling and capitalization matter when answer this question. We suggest asking the question as follows: (insert question here).

If you, as the instructor, do not have a GitHub account, you will need to create one as well. This student information will be needed to create your GitHub organization for your course.

## 5.2 GitHub organization

GitHub organizations are shared accounts where instructors and students can collaborate across many projects at once. Using your account, you can create a new GitHub organization by clicking on your profile icon in the upper right hand corner, go to *Settings*, *Access*, *New Organization*. It's suggested to name this organization the name of your class and the current semester you teaching in (i.e., sta199-s23). One of the many benefits of teaching through GitHub is the ability to re-use what you currently create as a template for subsequent semesters. Once your organization is created we can use packages within R and R-studio to efficiently invite students and create assessments for them.

## 5.3 Setting Up R & R-Studio

R is a statistical programming language for computing, modeling, and data visualization, while R-studio is an integrated development environment for R (cite). Both are freely available to download and use. In an introductory course, it is recommended to minimize student frustration and distraction through the use of pre-packaged computational infrastructure (Çetinkaya-Rundel and Rundel, 2018). Per this recommendation, STA199 has students use R and R-studio through a *Duke Container*. Duke containers provides instructors at Duke university the opportunity to facilitate the use of different software, such as R and R-studio, through an online container instead of needing students to locally download both programs. Additionally, instructors have the ability to manage and install packages students will need for the semester, helping provide a neat and well organized starting experience with a new statistical language. (insert discussion on how this is done).

(Thoughts: Transition to R-studio cloud discussion? What alternatives can instructors use to imitate Duke containers if this is something they do not have access to at their university.)

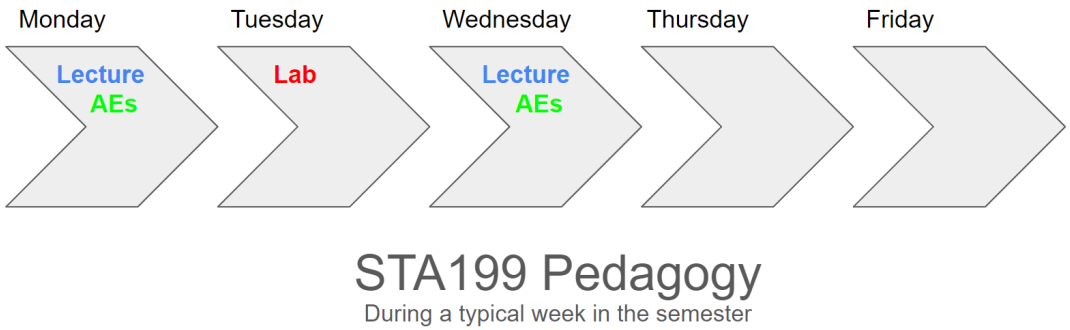
## 5.4 Using R & R-Studio to manage organization

(insert discussion / how-to on ghclass): includes invitations.

(set the stage for writing on how to create repos for students is subsequent sections *which will be a good segway*)

R, R-studio, and GitHub aids in the creation and implementation our data science pedagogy. This includes the creation and distribution of in-class AEs, lectures, labs, and assessments. In the following sections we describe in detail both how to create, structure, and distribute

materials used in STA199.




Unpack the image below

# 6 Pedagogy

– Reader should want to continue reading by thinking "so how do I teach this"

In this course, we have chosen a combination of teaching methods, interactive activities, and learning assessments to help best prepare introductory data science students the tools they need to be successful outside of university or in future coursework. Our pedagogy includes facilitating in-class AEs, facilitating lectures, running a lab, and assigning assessments to provide students an opportunity to show what they've learned.

## 6.1 Application Exercies (AEs)

A majority of the time in class will be dedicated to working on AEs. These exercises are live-coding opportunities to practice applying data science concepts and code introduced through preparation materials. AEs are no-stakes assessments, often graded on completion, where students have the ability to write and edit code while asking questions at the student or class level and receive immediate feedback.

– How GitHub and R are used to create AEs for students

It is up to the discretion of the instructor on the content that goes into the AEs. This can include having students write code themselves, fill in the blank coding exercises, commenting on complete code, or a combination of such questions.

– Kaplan Way (Practice) / Active learning implementation of AEs (typical class day strategies / discussion)

## 6.2 Lecture

A typical week yields 75-minute lectures on Mondays and Wednesdays. These lectures take up part of the class time, and are designed to introduce new concepts or review topics from the preparation videos in a more traditional format.

– Creating slides using Quarto in R (?)

– Kaplan Way (Prepare) / Active learning implementation during lecture (typical class day strategies / discussion)

Lectures are recorded and made available to students with an excused absence upon request.

## 6.3 Labs

– apply the concepts discussed in lecture to various data analysis scenarios, with a focus on the computation

– Individual and team based

– Repo creation

– Kaplan Way (Perform) (typical class day strategies / discussion)

## 6.4 Assessments

– Kaplan Way (Perform)

– HW

- apply what you've learned during lecture and lab to complete data analysis tasks

– Lab

- apply the concepts discussed in lecture to various data analysis scenarios, with a focus on the computation
- Individual and team based
- Repo creation

– Exams

- opportunity to demonstrate what you've learned in the course thus far
- take home exams

– Project

- analyze an interesting data-driven research question
- group project
- describe the process / expectations of the project

$\widetilde{\frown}$ The writing above should be in a form where there is practical / useful information for the reader. If they were designing the course, do they better understand the assessment structure of an Intro to Data Science course?

# 7 Discussion

It is imperative that universities create and implement modernized data science curricula to both prepare and inspire students to continue their data science education. This starts with the design and development of an introductory data science course. At Duke University, the technological preparations students receive in STA199 have pushed higher level courses, such as courses in regression, to incorporate Quarto for reproducibility and GitHub for collaboration and version control.

However, there is little consensus on the technological and implementation procedures necessary to create and conduct a modernized introductory data science course. We believe that the *Introductory to Data Science and Statistical Reasoning* helps establish a start to a consensus while providing a flexible framework for other instructors to create and facilitate their own introductory data science course.

It is highly advised that live coding sessions be the main staple of your classroom when teaching students. Live coding sessions continue to be accepted as one of the best pedagogical practices for teaching coding (https://dl.acm.org/doi/abs/10.1145/3430665.3456382), and have been met in our classrooms with overwhelming positivity from students: "insert more quotes."

For this to successful, we suggest spending the first couple classes establishing a routine with your students. This routine ensures that students clone the live coding exercise (application exercise) prior to the start of class, and understand that the expectation is to "learn through doing" in live coding sessions.

Despite this suggestion, there are always situations where students can fall behind in class, limiting the amount of value they may receive from the day's lesson. Thus, we have come

up with strategies to try and mitigate the situation. First, the majority of student coding at the beginning of the semester is done through fill in the blank templates created by the instructor. This tends to ease tension for those first learning code, and help instill confidence when the code runs. Secondly, it is suggested to design questions in such a way where there is little dependency across questions. This means that, if a student falls behind and is unable to answer one question, this will not deter them from being involved in upcoming questions. If you do have questions dependent on each other, we advise providing the answers to the beginning questions in the same or different document for students to reference and run so they can continue following along.

With the time investment needed to create enticing and interactive AEs, it is critical that these resources are easily transferable across semesters. The design and facilitation of this course is through GitHub. This ensures that the course content is feasible to adapt to teach for multiple semesters as classes change and data science continues to evolve. In short, there is value in the investment to quality resources early in the classes development, saving time for future renditions.

Additional benefits in deciding to run the course through GitHub help alleviate the burdens of large class sizes. Using the `ghclass` package in R, an instructor can seamlessly distribute homework, labs, and live coding exercises to as many students as necessary in little time.

# 8 Supplementary materials

– Pros (What works well)

Ex. Live coding good

– Cons (What could be improved)

Ex. Students can fall behind during live coding sessions

– Experiences

Examples include

– Feasible to adopt to teach for multiple semesters as data science continues to evolve. There is value in the investment.

– Computational Resources

– Human Resources

the *Kaplan Way*. The Kaplan Way is a learning model "that combines a scientific, evidence-based design philosophy with a straightforward educational approach to learning" (Schweser 2023). This learning model posits a three-phase learning strategy: Prepare, Practice, and Perform **TO DO: Check in with Maria about provenance of PPP**. During each of these phases, students are equipped with the appropriate tools to acquire knowledge, be given an opportunity to apply what they know, and to demonstrate mastery of the tasks at hand.

# References


Anderson, P. E., Bowring, J. F., McCauley, R. A., Pothering, G. J. & Starr, C. W. (2014), 'An undergraduate degree in data science: curriculum and a decade of implementation experience', *Proceedings of the 45th ACM technical symposium on Computer science education* .

Asamoah, D., Doran, D. & Schiller, S. (2015), Teaching the foundations of data science: An interdisciplinary approach.

Beckman, M. D., Çetinkaya Rundel, M., Horton, N. J., Rundel, C. W., Sullivan, A. J. & Tackett, M. (2021), 'Implementing version control with git and github as a learning objective in statistics and data science courses', *Journal of Statistics and Data Science Education* **29**(sup1), S132–S144.

**URL:** *https://doi.org/10.1080/10691898.2020.1848485*

BLS (2022), 'Occupational outlook handbook'.

**URL:** *https://www.bls.gov/ooh/math/data-scientists.htm*

Danyluk, A., Leidig, P., McGettrick, A., Cassel, L., Doyle, M., Servin, C., Schmitt, K.

& Stefik, A. (2021), Computing competencies for undergraduate data science programs: An acm task force final report, *in* 'Proceedings of the 52nd ACM Technical Symposium on Computer Science Education', SIGCSE '21, Association for Computing Machinery, New York, NY, USA, p. 1119–1120.

**URL:** *https://doi.org/10.1145/3408877.3432586*

De Veaux, R. D., Agarwal, M., Averett, M., Baumer, B. S., Bray, A., Bressoud, T. C., Bryant, L., Cheng, L. Z., Francis, A., Gould, R., Kim, A. Y., Kretchmar, M., Lu, Q., Moskol, A., Nolan, D., Pelayo, R., Raleigh, S., Sethi, R. J., Sondjaja, M., Tiruviluamala, N., Uhlig, P. X., Washington, T. M., Wesley, C. L., White, D. & Ye, P. (2017), 'Curriculum guidelines for undergraduate programs in data science', *Annual Review of Statistics and Its Application* **4**(1), 15–30.

**URL:** *https://doi.org/10.1146/annurev-statistics-060116-053930*

Dogucu, M. & Çetinkaya Rundel, M. (2022), 'Tools and recommendations for reproducible teaching', *Journal of Statistics and Data Science Education* **30**(3), 251–260.

**URL:** *https://doi.org/10.1080/26939169.2022.2138645*

Donoho, D. (2017*a*), '50 years of data science', *Journal of Computational and Graphical Statistics* **26**(4), 745–766.

**URL:** *https://doi.org/10.1080/10618600.2017.1384734*

Donoho, D. (2017*b*), '50 years of data science', *Journal of Computational and Graphical Statistics* **26**(4), 745–766.

**URL:** *https://doi.org/10.1080/10618600.2017.1384734*

Kokkelenberg, E. C., Dillon, M. & Christy, S. M. (2008), 'The effects of class size on student grades at a public university', *Economics of Education Review* **27**(2), 221–233.

**URL:** *https://www.sciencedirect.com/science/article/pii/S0272775707000271*

Redmond, F. (2022), With a rise in computing disciplines comes a greater choice of computing degrees in higher education, *in* 'Proceedings of the 22nd Koli Calling International Conference on Computing Education Research', Koli Calling '22, Association for Computing Machinery, New York, NY, USA.

**URL:** *https://doi.org/10.1145/3564721.3565946*

Schwab-McCoy, A., Baker, C. M. & Gasper, R. E. (2021), 'Data science in 2020: Computing, curricula, and challenges for the next 10 years', *Journal of Statistics and Data Science Education* **29**(sup1), S40–S50.

**URL:** *https://doi.org/10.1080/10691898.2020.1851159*

Schweser, K. (2023), 'Our philosophy'.

**URL:** *https://www.schweser.com/about-kaplan/philosophy*

Çetinkaya Rundel, M., Dogucu, M. & Rummerfield, W. (2022), '5ws and 1h of term projects in the introductory data science classroom', *STATISTICS EDUCATION RESEARCH JOURNAL* .