

## NBTEACH: SIMPLE WORKFLOWS FOR TEACHING WITH JUPYTER

Adam Blake<sup>1</sup>, Benjamin Winjum<sup>2</sup>, and James W. Stigler<sup>1</sup>

<sup>1</sup>UCLA Psychology Department

<sup>2</sup>UCLA Office of Advanced Research Computing

Los Angeles, CA 90095-1563

adamblake@g.ucla.edu

*Jupyter notebooks have great potential to facilitate the teaching of statistics and data science. However, setting up Jupyter for classroom use is intimidating for many college and high school instructors. nbteach is an extension for Jupyter and JupyterHub designed to make it easy for instructors to create, share, collect, and grade Jupyter notebooks in the context of statistics and data science education. While other solutions exist, nbteach is unique in its focus on the needs of the non-computer-science-savvy community. nbteach is currently available to students and their instructors free of charge as a cloud service at [nbteach.io](https://nbteach.io).*

### BACKGROUND

Over the past several years our research group at UCLA has been developing, implementing, and continuously improving an online interactive textbook for teaching introductory statistics and data science. The book includes more than 1200 formative assessment questions and R programming exercises, and takes a computational approach to the development of core concepts (Stigler et al., 2020; Son et al., 2020). The current version of the basic textbook is available at <https://coursekata.org>.

We started at the college and university level, but recently have expanded the program into high schools. In order to adapt to the remote learning contexts necessitated by the COVID-19 pandemic, and also as a means of adapting the materials across a wide variety of disciplinary contexts and educational levels, we have been using Jupyter notebooks to structure in-class interactions. A Jupyter notebook can function as both a lesson plan and as a worksheet (of sorts), giving students something to work on during class, and JupyterHub enables us to provide every student with an easily accessible and common set of computational resources and pre-defined Python and R environments. For assessments, notebooks can be structured to tackle a real-world problem, and with some help from the tools described later in this paper, these genuine, project-based assessments can be used in large-scale classes (we used them in 200+ student classes). Importantly, notebooks have the capacity to not only capture students' work on data science projects, but can also provide an insight into their thinking. For example, compare a series of exercises on single sample t-tests to a full notebook examining evidence for an age-discrimination case visually, verbally, and statistically.

We started looking for an easy way to publish notebooks to a class of students, give students a way to submit their completed notebooks back to the instructor, and then review students' work for purposes of grading their progress. We initially used nbgrader (Blank et al., 2019), but found it to be over-complicated for our purposes. Although nbgrader includes excellent tools for autograding students' code submissions, our teaching of statistics and data science does not so much require autograding of *code* as it does a way to examine students' *analyses and interpretations* of results. For example, students using nbgrader write functions that will transform data to desired outcomes and then to test the output of those functions. However, because the students we are targeting are seeking to learn statistics and data science, we avoid teaching concepts that many programming-oriented curricula take for granted, like writing functions. That is, we teach students to use R as a tool to use in their analyses, but we are not teaching them to program with R. We developed nbteach ([nbteach.io](https://nbteach.io)) as an alternative, and we built in several tools to make teaching and grading simpler for such use cases.

### TECHNICAL OVERVIEW

The target audience for this paper is educators who do not want to set up their own JupyterHub but do wish to use nbteach in their classes. This section provides a brief overview of the components of nbteach.

nbteach is available as a cloud service at [nbteach.io](https://nbteach.io). It is implemented as an extension to both the Jupyter Notebook and JupyterHub technologies ([jupyter.org](https://jupyter.org)). It has three main components: a modified and extended JupyterHub, which does all of the behind-the-scenes work like user management;

an extension to the Jupyter Notebook, which alters the user-interface and user-experience depending on role and context (e.g. a student working on classwork, or an instructor authoring an assignment); and a dashboard, which provides an organized and simple access point for students and instructors to access published and submitted notebooks. These three components are connected via an API exposed on the JupyterHub.

This system was originally developed specifically to work within the free online courses hosted at CourseKata ([coursekata.org](https://coursekata.org)). In those courses, the dashboard, created as a page within the user's learning management system (LMS), was the main access point to the user's Jupyter server. The management of notebooks assigned to and submitted by students was accomplished using the dashboard, which was updated using API calls to and from the nbteach extension. In the new freestanding version, a dashboard has been built as a separate web application to provide the interface that formerly was accessible only within the LMS. We currently provide this dashboard application as part of a hosted cloud service – not only to manage the distribution and collection of notebooks, but also to manage functionality previously provided by the LMS, including the ability for teachers and students to register for accounts and for teachers to assign students to classes.

When in a notebook, the front-end extension provides enhancements to the user experience. Instructors gain the ability to author and publish notebooks to their classes. When authoring these notebooks, instructors can make a number of modifications like locking cells, providing a submit button, or rendering interactive multiple-choice questions. When a student opens their copy of the notebook published by the instructor, these tools will automatically be displayed and run. The full scope of these tools is demonstrated in the workflows in the next sections.

Due to the nature of some of the modifications we have made to the JupyterHub, installing the entire nbteach system can be cumbersome. We are working to improve this process and possibly make it easier to download and install on a server of your choosing. At present, we are offering access to our nbteach installation, which runs on the Google Cloud Platform.

## BASIC WORKFLOW

We start our description of nbteach with a simple workflow teachers would use for authoring, distributing, and grading notebooks in one or more classes of students. In the next section we will introduce more advanced cell-level authoring tools.

The teacher starts by opening a Jupyter notebook hosted on an nbteach-enabled version of JupyterHub. This notebook can be a new, empty notebook or they can upload one that they already have. In addition to the normal Jupyter interface, the main toolbar will have an “nb” icon with four new buttons: Preview, Reset, Class Tools, and Publish.

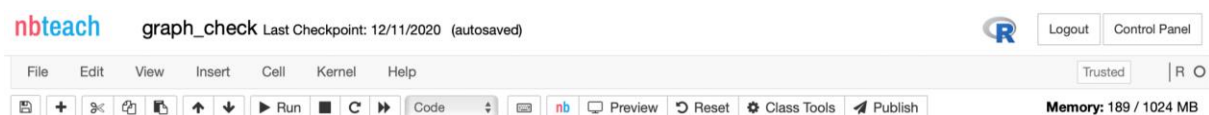


Figure 1. The main Jupyter toolbar with nbteach actions added to it

To distribute the notebook to their students, the teacher clicks the Publish button. This will open a dialog box so that they can select the classes they want to publish to and set any notebook-level options. At the most basic level, the teacher checks the box next to the classes to which they want to publish the notebook, and then clicks Publish at the bottom of the dialog box. This will create copies of the notebook for each student and instructor in the class, which are immediately accessible from their own account in the appropriately named “classwork” folder (also available via the dashboard at [nbteach.io](https://nbteach.io)). These classwork copies are the notebooks that students can work on, and which teachers can use in guided lectures or other such instruction.

After the notebook is published, the master copy is locked so that it cannot be edited, deleted, or moved. This ensures that there is always a source of “truth” for what the published notebook looked like, and serves as a place to manage the notebook. If edits need to be made, the teacher can Unpublish

the notebook—which removes it from each student’s account—and then make the edits and Publish it again.

*Option: Add a submit button*

Although the workflow above can be used to distribute a notebook to students, most instructors will want to provide a way for students to submit their completed work back to the teacher. When publishing the notebook, the instructor can check an option to add a submit button to the toolbar when the notebook is opened by students. When a student clicks that button in their notebook, a copy of their current notebook is sent to the teacher’s account and saved there. Students can submit as many times as they like, with each submission overwriting the last. Submissions are allowed as long as the teacher leaves them “open”, and they can be “closed” via a button on the dashboard or in the master or classwork copies of the notebook.

The submitted notebooks are saved in a read-only state on the instructor’s account, and they are named like `classwork/[class name]/[notebook name]/[student email]_[notebook name]`. These notebooks are also accessible directly from the dashboard, where they also can be downloaded, either individually or as a batch for the entire class.

*Option: Lock the notebook*

A second option available at time of publishing is to *Lock the notebook so students can’t add or delete cells*. In short, students will often accidentally delete important cells that have prompts or information they need, or add cells making the notebook more variable and harder to grade. Usually they aren’t trying to get their notebooks all messed up like this, so you might want to prevent them from doing so!

Locking the notebook will prevent the structure of the notebook from changing. Not only are actions like inserting, cutting, deleting, and moving cells prevented, their buttons and menu items are also removed from the toolbar. This feature is especially useful when using notebooks for quizzes or exams because it reduces the search space when a student is trying to perform other actions. Combined with the specialized cell types described below, locking also ensures that their notebook has a predictable form and structure, which can greatly benefit the grading process.

## SPECIALIZED CELL TYPES

One challenge of working with novice users is that Jupyter notebooks are very easy to mess up. If you are in Command Mode, a few keystrokes can add multiple cells, delete others, and open dialog windows. To have better control over what students do (and don’t do!), we have created a set of cell-editing tools that allow the teacher to specify the behavior of each individual cell.

When nbteach is enabled, each cell will have its own dropdown menu. In the default state, this menu will display three dashes (---). These are regular Jupyter cells, and like normal Jupyter cells, they can easily be changed from Markdown to Code cells and back again. At any time, the instructor can revert all cells back into normal Jupyter cells by pushing the Reset button on the nbteach toolbar at the top of the notebook.

*The Nine Special Cell Types of nbteach*

In addition to normal Jupyter cells (---), nbteach allows the instructor to set each cell to one of 9 special types: 3 types of Code cells, 3 types of Markdown cells, and 3 new cell types called Response cells. The main distinctions between the different cell types is in what students are allowed to do with the cell. Once a cell is set to one of the 9 special types, students will not be able to delete the cell. If a cell is a Code cell the student cannot change it to a Markdown cell, and vice versa. Depending on the cell type, the instructor can specify whether students can edit the cells, run the cells, or just view them. The 9 types of cells, with their associated permissions, are summarized in the table below.

Table 1. The different types of cells you can make in nbteach and whether you can perform certain actions on those cells (yes = ✓, no = ✗)

Cell Type	Delete	Edit	Run
(---) Normal; default	✓	✓	✓
Code: Edit-And-Run	✗	✓	✓
Code: Run-Only	✗	✗	✓
Code: View-Only	✗	✗	✗
Markdown: Edit-And-Render	✗	✓	✓
Markdown: Render-Only	✗	✗	✓
Markdown: View-Only	✗	✗	✗
Response: Markdown	✗	✓	✓
Response: Code	✗	✓	✓
Response: Multiple-Choice	✗	✗	✗

### *Response cells*

Perhaps the most useful cell types for teachers are the Response cells. Response cells are especially useful when using Jupyter notebooks for student assessment and grading. There are three types of response cells: Markdown, Code, and Multiple-Choice. Once a cell is set as a response cell, a light-colored border appears around the cell. This helps students see exactly where they are expected to submit their work, be it code or text in answer to a challenge or question.

Multiple-Choice cells allow the teacher to embed multiple choice questions into the notebook. To author a multiple-choice question, simply select Response: Multiple-Choice from the cell dropdown menu. Then, write the question in normal markdown format, formatting the response options as a bulleted list (with each choice preceded by a hyphen or asterisk and a space):

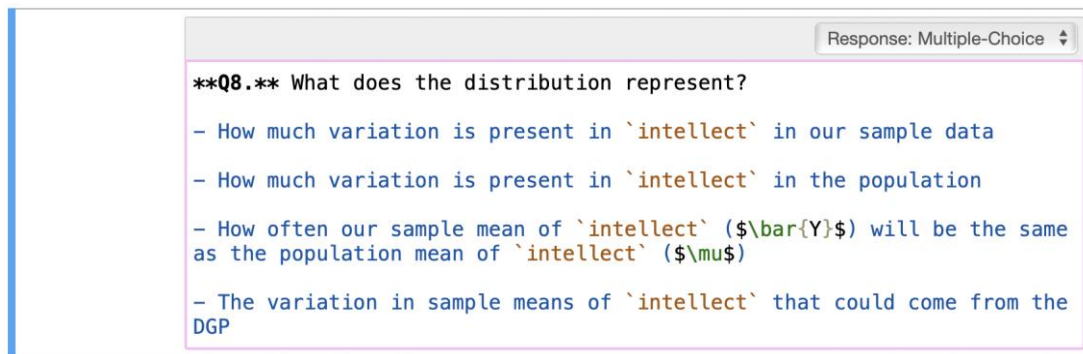


Figure 2. A multiple-choice cell in authoring mode (before being rendered to an interactive item)

When you render the cell, it will look like this:

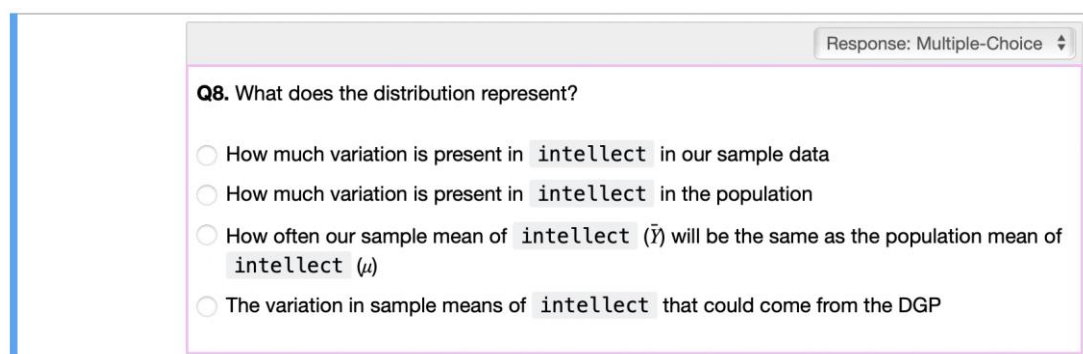


Figure 3. The same multiple-choice cell as in Figure 2, but rendered to its interactive form

## GRADING NOTEBOOKS

nbteach was designed for instructors with different grading needs and programming literacies. It is a non-trivial task to teach students to write code that can be graded with instructor-written test code, and an even more difficult task to teach instructors how to write that test code. Many instructors at the High School and Community College level who would like to adopt new technologies like Jupyter are faced with time constraints that prevent them from exploring new options. Keeping these instructors in mind, we included some tools that made the transition to using Jupyter Notebooks for assessments a relatively smooth process.

Notebooks submitted by students can be downloaded from the nbteach dashboard and graded individually. This may work well for small groups of students or free-form projects, but can be a long process similar to grading essays. When instructors utilize the Response cells described above, they can also use a simple tool we provide on the dashboard that will extract all student responses into a CSV file—one column per response cell—that can be opened in Excel or similar spreadsheet software. This makes it much easier to grade students' written analyses, multiple-choice responses, and even their code (though, if needed the output of that code can be checked by opening the student's submission or looking at the static version of the submission from the dashboard). In nbteach grade sheets, students' responses to multiple-choice questions are in an easy to grade numeric form, where 1 corresponds to the first option, 2 to the second option, and so on.

One thing we have learned from experience: If an instructor plans to grade the notebooks (e.g., for a quiz or exam) we highly recommend choosing the option to *Lock the notebook* when publishing it. Students will often create a new code cell and try to enter markdown text (or vice versa), or write their answers into new, non-Response cells which will not be included in the downloadable grade sheet.

Locking the notebook will ensure that students enter their responses in the response cells created by the teacher instead of creating their own new cells.

## EMBEDDING IN-CLASS TOOLS IN THE NOTEBOOK

Finally, nbteach allows the teacher to embed in-class tools into the notebook, a feature we have found especially useful for remote instruction during the pandemic. Any web app or tool that can be embedded with a URL in an iframe can be added to an nbteach notebook.

For example, we set up a simple Google Form called *Ask for Me* that allows students to post a question for the teacher by entering it into a pop-up window that opens when the student clicks an icon in the lower left-hand corner of a notebook.

Setting up in-class tools is accomplished by clicking the Class Tools button on the teacher's control bar at the top of the notebook. Configuring a tool is as simple as entering a URL for the tool, then selecting an icon to indicate to students how to open the tool. Once configured, the teacher can add any number of their configured tools to a notebook at the time of publishing.

We have tried various in-class tools so far: Poll Everywhere (for delivering polls during class); a web-based chat application; Google Docs, to share student ideas; and a homemade cheat sheet we created to show students some of the most useful keystrokes for Jupyter. Undoubtedly, others will discover many kinds of apps that can be included in notebooks.

## CONCLUSION

Because Jupyter extensions are often developed by people who feel comfortable with programming (e.g., computer scientists), they may not be easily accessible to teachers who are not comfortable with tasks like writing functions. It is daunting for these teachers to teach students to write gradable functions, and to write functions that grade functions. With nbteach, we hope to provide an on-ramp for teachers who may want to use R or Python in their statistics or data science courses, but are discouraged by the learning curve required to use many of the available tools. Our goal in developing nbteach was to provide a simpler platform that would empower instructors who would shy away from tools that can greatly benefit their students.

nbteach is currently being used by thousands of students in colleges, universities, and high schools, and we are continually working to improve the student and teacher experience. Though most of these users are using nbteach in tandem with the *CourseKata Statistics and Data Science* course, we wanted to make it available to the wider community. We are offering the system for free to any educators that would like to try it out in their classrooms (we may develop a pricing scheme in the future to recoup server costs as the user base continues to grow). Just visit [nbteach.io](https://nbteach.io) or email the corresponding author for more information.

## REFERENCES

- Blank, D. S., Bourgin, D., Brown, A., Bussonnier, M., Frederic, J., Granger, B., ... & Willing, C. (2019). nbgrader: A tool for creating and grading assignments in the Jupyter Notebook. *The Journal of Open Source Education*, 2(11).
- Son, J. Y., Blake, A. B., Fries, L., & Stigler, J. W. (2020). Modeling First: Applying Learning Science to the Teaching of Introductory Statistics. *Journal of Statistics Education*, 1-23.
- Stigler, J. W., Son, J. Y., Givvin, K. B., Blake, A., Fries, L., Shaw, S. T., & Tucker, M. C. (2020). The better book approach for education research and development. *Teachers College Record*, 123(2).