

Enter our title here

Tessa McDonnell, Shannon Pileggi,

Statistics Department, California Polytechnic State University, San Luis Obispo

1. Introduction

It is no surprise that technology plays a vital role in practicing statistics and as the technological revolution continues to unfold, the use of statistical software has become increasingly emphasized in education. By integrating the capabilities of technology in introductory statistics courses, students can explore fundamental concepts by analyzing and interpreting real data sets (?), giving them an authentic view of how statistics is practiced outside of the classroom (?). The revised 2016 *Guidelines for Assessment and Instruction in Statistics Education* (GAISE) College Report has also prompted the use of more technology in introductory statistics courses, encouraging educators to use technology "as a way to explore conceptual ideas and enhance student learning" (AmericanStatisticalAssociation2016).

The goal for any introductory statistics course is to help students develop the ability to think statistically (?). Introducing technology into the curriculum can greatly strengthen this ability by illustrating statistical concepts such as variability, randomness and hypothesis testing. Although numerous statistical technologies exist, *R* and *R Studio* provide extensive statistical capabilities (free of charge) and are commonly used in statistics courses (?). *R* allows students to explore real-world data sets which can not only enhance their understanding of concepts, but can also spark an interest in the domain of statistics (?).

With this increased use of *R* in classrooms, there is a need for an effective way to get students familiar with *R*. While *R* software provides users with extraordinary abilities there is no doubt a steep learning curve. To facilitate this transition to *R* programming, popular interactive teaching technologies have been created including Swirl (need to cite this?), Learn*R* and Try*R*. These tools allow students to work through coding problems while receiving immediate feedback in and out of the classroom. Swirl is an open source *R* software package that allows students to learn *R* programming in their *R* console. This interactive learning tool contains several pre-existing lessons but what makes Swirl so popular is its versatility; educators can create their own lesson (or tailor a pre-existing one) to fit their curriculum and assign it to students (?). While swirl can be linked to a Massive Open Online Courses (MOOCs) like Coursera through API, this is not available for the classroom learning management systems (LMS).

*I'm confused about Learn*R* and think someone else should write about it.*

Try*R* (Code School 2014) is a web-based program and does not have this reproducible capability. Since public content authoring is not available, educators are limited on what they can teach their students. The *R* console can be intimidating to introductory statistics students

so an advantage of a web-based teaching tool is the clean and easy to use interface. This is an important quality given that a lot of the time students struggle more with software than the statistical concepts themselves (?). When students only need their web browser to participate, all installation complications are avoided and valuable class time can be focused on the statistics, not computer issues.

DataCamp has the same reproducible capabilities as Swirl but is a web-based application; educators can create their own courses that coincide with their lessons and students can access the course without having to install R. Courses are available through the DataCamp website and students can access the lesson through their web-browser on their laptop, tablet or mobile device, making the courses accessible to anyone with an internet connection. The DataCamp website offers pre-existing free community courses and specialized 'Premium' classes that you need to purchase. However, when used for academic purposes the premium DataCamp content is also free of charge.. (i'll come back to this). Any college educator can create a classroom group on the DataCamp website, add student's to the group, assign pre-existing 'community' courses and assign a course that they've created for the particular class.

The collection of community courses available at DataCamp include a wide range of topics from 'Introduction to R' to 'Quantitative Risk Management in R' so novice through advanced programming students will benefit. DataCamp can be used in the classroom in a number of ways: as traditional homework assignments, as an activity during lab periods, as pre-class assignments for a flipped classroom, and even as a test or assessment tool.

A feature that sets DataCamp apart from competing technologies is the ability to track student progress. Educators can set deadlines for assignments and easily see not only when students complete an assignment but also which sections the students had trouble with in a detailed report (datacamp/teach website). Student scores can be automatically updated to a current gradebook when you link the schools' learning management system (LMS) to DataCamp, allowing for a smooth integration of student work.

Each chapter in a DataCamp course is made up of modular exercises. Modularity is critical component in designing a statistical learning tool because it allows independent pieces to be easily added, modified or removed from the entire functionality (?). Course exercises can be added, modified, or deleted as new topics arise and instructors can tailor these exercises to meet the needs of a particular class. The only disadvantage of the modularity feature is that the sessions are not cumulative; every exercise begins with a clear workspace so an object defined in a previous exercise needs to be defined again. Each of these exercises can be given a number of redeemable points which the students can see when they begin the exercise, introducing a gamification feature to the learning experience. If the student completes the exercise without any help they will receive the full points. If they ask for a hint, 30 percent of the available points will be deducted and all of the points will be deducted if they ask for the entire answer. When in a classroom group, students can view their progress (points and chapters completed) as well as then progress of their peers. This can help students perform better because they want to be on par with the rest of the class; furthermore, when students can see the number of available points for an exercise it will motivate them to meet their personal achievement goals. (?).

2. How to Use DataCamp

2.1. Pre-existing R courses

DataCamp has a range of available free community courses that educators can utilize for supplemental R practice.

2.2. Creating a DataCamp Course

To create a course on DataCamp you will first need to create a DataCamp account by visiting www.datacamp.com/teach and also a GitHub account. The simplest way to begin building the course is to use the template course files from the 'Create DataCamp Course' dialog. This will automatically create a GitHub repository that is linked to DataCamp; any changes that are made through DataCamp will update the GitHub repository and visa versa. The GitHub repository will contain three files: README.md, course.yml, and chapter1.Rmd. Readme.md contains helpful information of how to get started, citing resources to learn more about the process. course.yml is a file that contains general information about the course you're creating: title, university, description, difficulty level, ect. chapter1.Rmd is a template chapter file with examples of 'normal' interactive coding exercises and multiple choice questions. To add more chapters to an R course, create new files in the repository and name them chapter1.Rmd, chapter2.Rmd, chapter3.Rmd ect.

Educators are able to edit their courses in a number of ways but the 'Teach Editor' located in the teach dashboard is the most straight forward editor. The 'Teach Editor' is specifically designed to make the creation process of a DataCamp course as painless as possible, allowing writers to edit, preview, save changes and synchronize to a GitHub repository. Editing can also be done through GitHub's online editor and through git locally but these methods do not allow for automatic content viewing.

When starting to create a DataCamp course, it is helpful to view code from pre-existing community courses which can be found on GitHub. You can use these files as a guide to build your own course or duplicate the repository and tailor the content to fit specific needs. The DataCamp github repository contains links to the code for available courses; one of the most helpful when starting to build a course is 'datacamp/courses-intro-r' which contains the source files for the 'Introduction to R' course.

DataCamp courses may contain 3 different types of exercises: Normal, Multiple choice, and video.

A 'NormalExercise' is an interactive coding exercise containing a lesson section, instructions, pre-exercise code, sample code, hint, solution and a submission correctness testing coded (SCT). However, the interface for students (or the 'Preview mode') only includes the lesson, Instructions, R script panel and a R console panel.

The submission correctness testing code (or SCT) is the backbone for DataCamp's interactive

nature. If a student submits an incorrect answer, they will receive immediate feedback on what they are doing wrong and the system will guide them to the right answer. Instructors can use the default feedback that DataCamp automatically generates or write their own custom messages. The default feedback is meaningful and specific; most of the time custom messages are not necessary unless there is a common problem that students are struggling with in which more unique feedback is necessary. DataCamp provides a helpful R package called 'testwhat' which acts as a guide to writing and using SCTs (this can be accessed on DataCamp's GitHub repository: 'datacamp/testwhat'). 'testwhat' contains functions that can test different types of problems including loops, object definitions, printed output, functions, etc (datacamp website), this submission correctness testing code compares the ideal answer (from the solution) to the student's answer. Different arguments can be added to the test functions to alter the testing process and automatic feedback.

In the 'assignment' portion of the exercise, instructors can add additional information about the assignment that will help students understand and complete the instructions. The 'Instructions' section will contain the actual task the student is asked to execute. The 'pre-exercise code' sets up the workspace for the student; this section will contain code to import data sets, install packages and any other operation that you want to be in the workspace but hidden from students. The 'sample code' section contains code or notes that students view in the R script panel. Students also use the R script panel to write the code that solves the tasks in the instructions. If students are not able to complete the task without help, they can click on the 'Hint' button or eventually the 'solution'. When the student clicks the 'Submit Code' button, their answer passes through the SCT code where the code is checked for errors.

The 'MultipleChoiceExercise', as you would expect, contains a question and a list of various answers. Instructors write the questions in the 'lesson' portion of the exercise and offer solutions in the 'Instruction' section. Like the 'normal' coding exercises, the lesson, instructions, pre-exercise code and SCTs have to be specified but sample and solution code do not. DataCamp offers two types of multiple choice questions, simple and regular. The simple multiple choice exercises are used for questions that do not require the R console or any R output. Regular multiple choice exercises allow instructors to pre-program objects or plots that the student needs to access to answer the question. Both types of multiple choice exercises are straightforward to write and can act as quick checks for understanding throughout the chapter.

Video exercises act as an instructional video addition to a DataCamp course; instructors can add short videos of them explaining concepts and with relevant slides that appear in the background. The DataCamp website https://www.datacamp.com/teach/documentation#tab_code_exercises provides more detail on the content of the various exercises.