Analysis of Application:

Introduction to 'R' – Programming, Wrangling and Visualization Basics

James Hardaway

College of Education, NC State University

ECI 517: Theoretical Foundations of Advanced Learning Environments

Dr. McKeown

November 12, 2021

Introduction to 'R' – Programming, Wrangling and Visualization Basics

My learning activity analysis focuses on an early lesson in ECI 586, An Introduction to Learning Analytics (LA). This course to fulfills one of my four class requirements to earn the Graduate Certificate in Learning Analytics from NC State's College of Education. While I am fairly new to the coding world, I have some experience in data and analytics. This paper will describe my experience completing an early learning activity for my first coding course. My goal is to apply the foundational theories of learning and instruction to evaluate the course's effectiveness while also developing recommendations that could improve learning outcomes.

Why Learning Analytics?

This introductory course prepares education professionals to collect and analyze data to "better understand and ultimately improve student learning and the contexts in which learning occurs. More specifically, this course provides an overview of Learning Analytics as a discipline, examples of its use in educational contexts, and applied experience with widely adopted tools and techniques used by education researchers and practitioners." (Kellogg, 2021a)

As a distance education course, content is primarily presented as asynchronous activities using NC State's learning management system, Moodle, as well as a variety of other online tools to present videos and encourage class discussion. While weekly opportunities exist for synchronous problem-solving and instructor office hours, the course is designed to be at the student's own pace. The course material provide via Moodle is augmented by each student having access to external instructional resources through Data Camp, RStudio Cloud, and LinkedIn Learning, to name just a few. In addition to web-enabled applications, students are encouraged to download a desktop version of R and RStudio to enhance independent practice.

This course uses a 3-part strategy to integrate the learning of new concepts with exercises and practice activities. Each new concept is introduced through the class text or articles. Once explained, the concepts are then explained in a practical sense through tutorials and primers where the students learn step-by-step how to code in R to produce products based on what was just learned. Finally, the conceptual learning is completed through a case study activity where the student applies everything learned during that lesson while developing a contextual product.

Unit 1 Tutorials: R Programming, Wrangling, & Visualization Basics

Unit 1 of the course is designed as a general survey of the data analytics field as well as how those processes are applied in education. As referenced in the syllabus, the second week of each unit consists of tutorials for learning how specific R programming modules (called packages) are used to analyze educational data. The primary goal of these tutorials is to support familiarity and fluency with R syntax and key functions for data analysis.

For this introductory unit, the tutorials give students practice on understanding how to examine data through the five core phases of *prepare*, *wrangle*, *explore*, *model*, and *communicate*. This unit contains two tutorials to familiarize students with the programming and two exercises to impart practical context and reinforce learning. The specific lesson plan is captured in the Appendix.

Tutorial Learning Outcomes

The table below describes how the various learning objectives for each of the student activities align to both Gagne's and Bloom's learning outcomes.

Table 1 - Unit 1 Tutorial Objective Comparison

Activity Objectives		Gagne Learning Outcomes	Bloom Cognitive Outcomes
Tutorial 1 (Basics)	Run Functions	Verbal Information Intellectual Skills Motor Skills (typing)	Knowledge Comprehension Application
	Build Objects		
	Graphic Plots		
Tutorial 2 (Data)	Extract Values		
	Subset Tables		
	Calculate Stats		
	Derive Variables		
Exercise 1	Basic Skills	Cognitive Strategies, Attitudes, Motor Skills	Analysis, Synthesis, & Evaluation
Exercise 2	Data Skills		

Learning Inputs and Means

Each tutorial presented multiple desired learning outcomes. In the first primer, *Data Basics*, the text described how we would employ run functions, build objects, and begin making graphical plots of data we'd imported into the program. Those descriptions served to reinforce learning from the previous week's assigned reading while also previewing what would be practiced by writing snippets of code in the tutorial knowledge checks (see Figures 2 and 3, Appendix).

This technique satisfied the development of declarative knowledge as a precursor to procedural knowledge (Driscoll, 2004, p. 155). It activated what I had put into working memory the week prior as the text contained visual examples describing how the code was used in the context of an actual R script. While all this would be considered a verbal learning outcome by Gagne, it also aligns well with Bloom's cognitive outcomes of knowledge and comprehension (Driscoll, 2004, p. 358). Lastly, there was some rudimentary motor skills practice due to the need to learn how to replicate some of the special characters required to code in R. While it's still typing, understanding which keys and shortcuts are paired with specific functions requires practice.

Upon completion of the skills primers, the week's activities conclude with coding exercises in RStudio. For this week, there was an exercise tied to each of the two primers. RStudio is a web-enabled coding environment that allows the instructor to segregate the week's learning tasks into separate projects. Each student logs in and can complete the exercises independently. Once completed, the instructor can see each student's work to grade (see Figure 4, Appendix).

These exercises provide the opportunity to apply the lessons learned during the tutorials. Bloom would describe this as the point where learners develop cognitive strategies and choose how they internalize personal feelings and states of understanding, what he calls attitudes. Using slightly more clear language, Bloom refers to these phases of learning as analysis, synthesis, and evaluation. This is where I would break down all the individual concepts learned and then reassemble them to solve unique problems. As the coding language R has multiple pathways to achieve specific outcomes, I had the freedom to choose from numerous arrangements to achieve the exercises' directed end states.

Learning Theory

The design of these tutorials and practical exercises was primarily driven by cognitivist theories of learning development. From the perspective of information processing, the instructor designed the exercises to reinforce **working memory** and the transition of information from **declarative** to **procedural knowledge**. The bulk of the activity was well-scripted with ample supervision and guidance from the instructor. We explored each new concept in isolation as a way to **chunk** the information for easier storage in our working memory. Each coding drill in the tutorials acted as **rehearsals** that mimicked what we would do in the graded exercises. Interactional theories of cognitive development were reinforced by the social and group aspect of problem discussion that happened weekly. Additionally, we were introduced to concepts visually and then taught how to design that image with a language (R in this case). This sequencing mirrored Bruner's concept of **iconic** and **symbolic** representation of development (Driscoll, 2004, p. 232). Finally, situated cognition influenced learning outcomes as we were introduced to the wider R programming ecosystem through the various instructional platforms. These platforms allowed us the creativity to build and submit coding examples to the broader **community of practice** if we developed a unique way to solve difficult problems.

As this learning activity occurred initially in the intellectual space and practically in an online coding environment, I believe its epistemic roots are drawn from rationalism and pragmatism. The knowledge required to achieve the learning objectives was created through reason as none of us had any preconceived or innate sense of understanding. While there were some sensory experiences (mainly visible), those were just a small part of the learning experience. From a pragmatic viewpoint, our exercises allowed us to reason to achieve specific solutions that could be understood objectively by the rest of the class. In other words, the solutions weren't reliant on internally held experiences and interpretations.

Some key assumptions that probably drove the instructor to develop this activity in such a structured manner was that most of us are new to the data science world. We have developed little in the way of long-term memory or recall that can be used for this class. At the same time, the instructor could assume a heightened sense of student motivation as we chose to take this class as an elective for our graduate programs. This assumption would lead to a belief we would be highly motivated to make this course a priority and apply some rigor in trying to understand the new concepts.

Analysis and Evaluation of Learning Activity

My enjoyment and performance in the course make me believe the course is designed well to achieve the stated learning objectives. However, as a new learner to data science and coding, I sometimes feel like I'm on an island in a very unfamiliar ocean. Something that would assist in developing a little more confidence in learning the new concepts would be to develop groups in class or partner the students with an experienced mentor that could provide consistent feedback as new concepts are introduced. The instructor doesn't have the time to do this himself with every student, but an option would be to expand the community of practice concept to include a working partnership with other students who have been doing this for a while.

Another option to develop some pre-cursor knowledge for new students would be to develop short, pre-requisite exercises for students to complete before the course begins. Many of the resources we use in class are free and self-paced. I could envision doing those in a few weekends before the semester starts to get a jump on overarching terminology and concepts. This could speed the pace of recall during some of the initial lessons.

References

Driscoll, M. P. (2004). *Psychology of learning for instruction* (3rd ed.). Allyn and Bacon

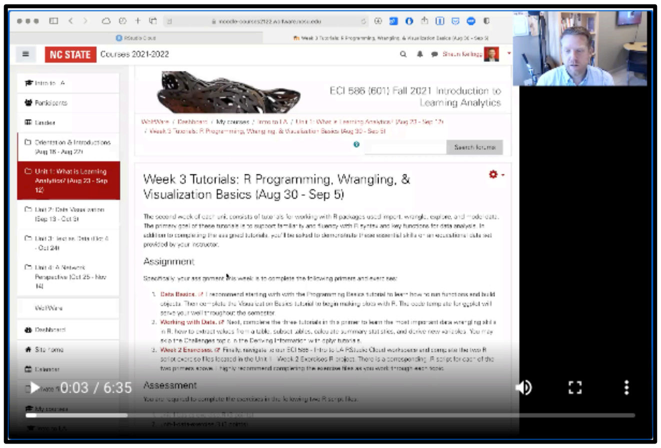
Kellogg, S. (2021a). *Syllabus*. An Introduction to Learning Analytics. Retrieved November 7, 2021, from https://sbkellogg.github.io/eci-586/.

Kellogg, S. (2021b). *Week 3 Tutorials: R Programming, Wrangling, & Visualization Basics*. ECI 586 (601) Fall 2021 Introduction to Learning Analytics. Retrieved November 7, 2021, from https://moodle-courses2122.wolfware.ncsu.edu/mod/forum/view.php?id=357620.

Appendix

Lesson Plan Materials for ECI 586 Week 3, *R Programming, Wrangling, & Visualization Basics* (Kellogg, 2021b)

Figure 1 - Assignment Overview Video



The second week of each unit consists of tutorials for working with R packages used import, wrangle, explore, and model data. The primary goal of these tutorials is to support familiarity and fluency with R syntax and key functions for data analysis. In addition to completing the assigned tutorials, you'll be asked to demonstrate these essential skills on an educational data set provided by your instructor.

Assignment

Specifically, your assignment this week is to complete the following primers and exercises:

1. **Data Basics.** I recommend starting with the *Programming Basics* tutorial to learn how to run functions and build objects. Then complete the *Visualization Basics* tutorial to begin making plots with R. The code template for *ggplot* will serve your well throughout the semester.
2. **Working with Data.** Next, complete the three tutorials in this primer to learn the most important data wrangling skills in R: how to extract values from a table, subset tables, calculate summary statistics, and derive new variables. You may skip the Challenges topic in the *Deriving Information* with *dplyr* tutorials.
3. **Week 2 Exercises.** Finally, navigate to our ECI 586 - Intro to LA RStudio Cloud workspace and complete the two R script exercise files located in the Unit 1 - Week 2 Exercises R project. There is a corresponding *.R* script for each of the two primers above. I highly recommend completing the exercise files as you work through each topic.

Assessment

You are required to complete the exercises in the following two R script files:

1. unit-1-basics-exercise.R (3 points)
2. unit-1-data-exercise.R (3 points)

These are not quizzes. **If you have questions about the tutorials or exercises, I encourage you to post to the forum below**, especially if others might benefit from the response to question. You are also more than welcome to respond to the questions posted by your peers. However, I do ask that you try work through the exercises independently before reaching out for support.

<u>Referenced Figures</u>

Figure 2 - Visualization Basics

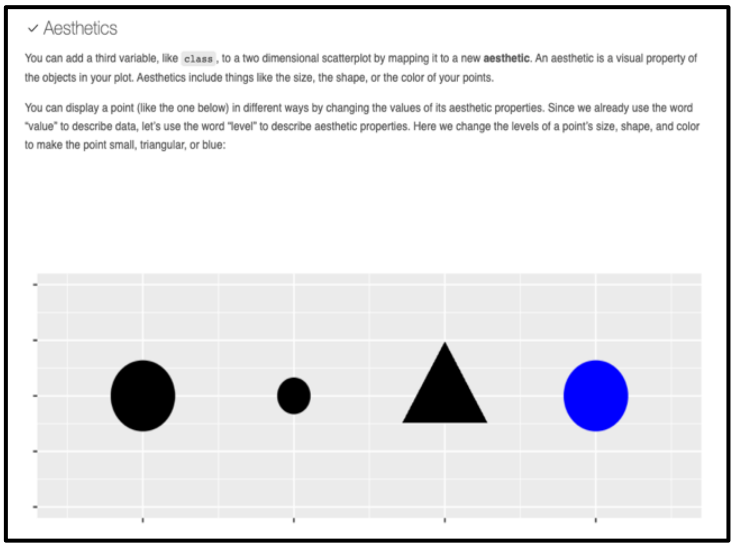
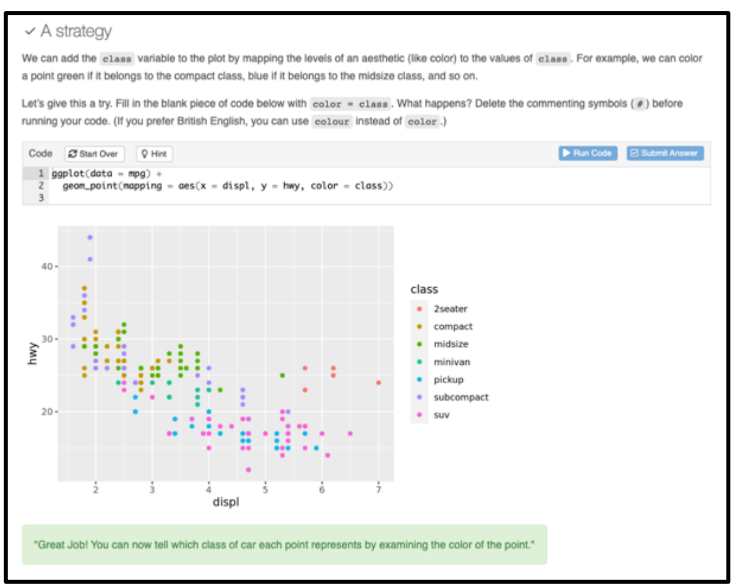


Figure 3 - Coding Snippet Example



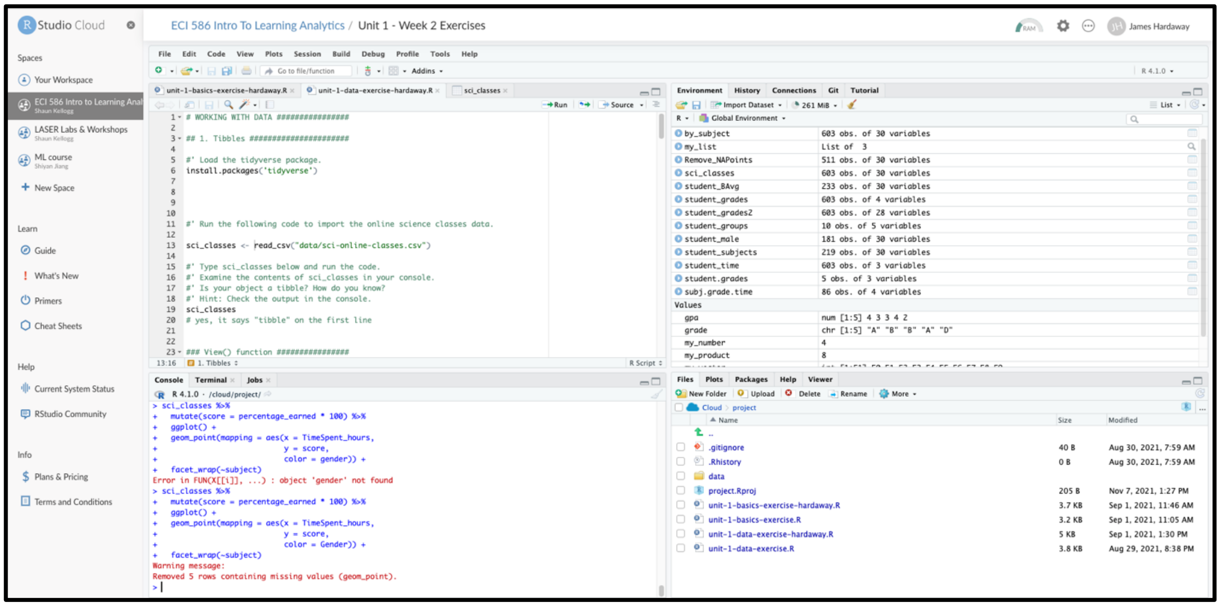


Figure 4 - RStudio Coding Environment