

**Applied Design Science Research for Coding Data Collecting Web Applications as Proof of
Concept with AI**

Michael Littrell, PhD, Faculty, College of Education and Human Sciences, Tennessee Tech
University

Andrea Arce-Trigatti, PhD, Faculty, College of Education and Human Sciences, Tennessee Tech
University

Abstract

Limitations of off-the-shelf software data collection platforms (e.g., Qualtrics, Survey Monkey) can possibly affect the types of survey data collected by higher education (HE) institutional data professionals. Overcoming this limitation could be beneficial for increasing the quality of results obtained when striving to support HE stakeholders. Consideration of creative and novel approaches to collect data in a way that overcomes common platform limitations has historically been constrained by steep software learning curves. R is a language often used at HE institutions for working with data. Generative AI (GAI) conversational coding paired with the R Shiny framework allows for the rapid prototyping of proof-of-concept (POC) web applications by HE professionals with limited software development experience, going beyond the limitations of existing platforms and possibly better supporting the specific contextual needs of HE stakeholders. Demonstrated in this work is a conversational coding example using a design science research framework applied to a hypothetical HE recruitment scenario. A novel survey data collection web application artifact is produced as an illustrative applied example to understand in what ways a conversationally coded app can extend the capabilities of HE data professionals beyond features offered by existing data collection platforms. The result is a functional POC and perhaps offers a fresh perspective on what is possible by HE data professionals with limited programming experience.

Introduction

Gathering survey data through data collection platforms (DCP) is ubiquitous and used throughout the world by higher education (HE) institutions. This data collection can be limited by the features of the platform (e.g., Qualtrics, Survey Monkey, Forms) possibly preventing HE professionals from serving stakeholders to the extent they could if these limitations did not exist. Generative AI (GAI) is a system that can create new content across a variety of outputs including pictures and text, changing the technical depth by which the typical HE professional who works with a computer can operate. GAI now allows end-users to consider more deeply the software they use even if they possess limited coding knowledge. When working with off-the-shelf DCPs, users tend to view the problem through the lens of the software features available to them which in turn can influence how they approach the process for gaining insights. The experience is then an on-rails one where question types and workflow options are accepted as the constraints of what can be accomplished for a given data collection effort, limiting what Tsai et al. (2019) underscore as potential personalized strategies to support quality improvement efforts. While these platforms are robust, this pre-determined process could prevent HE researchers from exploring creative and novel approaches to data collection that could yield better results such as more targeted and tailored insights (Agasisti et al., 2017; Tsai et al., 2019).

Purpose and Problem

The purpose of this Design Science Research (DSR)-informed work is to explore alternative pathways to considering electronic data collection through an example demonstrated creation of a proof-of-concept (POC) web application artifact for collecting higher education survey data. DSR is about solving problems to extend human knowledge by creating artifacts in an innovative way (Vom Brocke et al., 2020). Primarily, DSR is focused on pushing the

limitations of the capabilities of individuals and groups (Hevner et al., 2004). The approach utilized is meant to be a demonstration of how GAI can be used to fill in technical gaps by HE data professionals who may have limited experience with programming web applications.

The DSR framework is used to ground the example conversationally coded application in a hypothetical HE scenario so that this illustrative case is practical and applied for institutional professionals. The resulting POC will provide contrast to existing workflows found in off-the-shelf DCPs highlighting possibilities for extending predefined software features typically manipulated through the user interface (UI) of said applications. The work in this paper is meant to expose potential avenues of thought and tangible tools for overcoming the limited experiences offered in predefined survey platforms.

Research Questions

The research question for this study is focused on the resulting artifact and the expansion of knowledge which aligns with the DSR framework that will be used in the process. The question is:

Q1: In what ways can a proof-of-concept survey data collection application built with assistance from generative artificial intelligence extend the capabilities of institutional researchers beyond the limitations of existing data collection platforms?

Background and Context

Surveys in Higher Education

Survey research in HE is important for serving stakeholders and supporting institutional goals, but can be challenging to execute in a way that produces valid and reliable results for a myriad of reasons (Fosnacht et al., 2017). Adding to the challenge of collecting data, existing survey DCPs can be costly and on-rails encouraging the user to look at the data collection

problem through the lens of an existing set of tools available to them instead of a tailored medium that best supports the end goal. These capabilities are further limited by financial and organizational considerations that determine what type of DCP is available, and the flexibility of data-driven cultural norms at existing HE institutions (Agasisti et al., 2017; Tsai et al., 2019). Further, data-driven leadership strategies may dictate the level of institutional efficiency that can be reached regarding data collection capacities, as well as the professional development opportunities available to expand skillsets in these areas (Webber & Zhang, 2023).

Going beyond the constraints of the feature options within such an application can be challenging technically if it is even possible at all. DCP infrastructure must be continually renewed, with continued, available technological support necessary to help HE institutions adapt to changing data collection trends and demands, with few institutions able to address these financial and technical complexities (Webber & Zhang, 2023). To this end, scholars have attempted to address the limitations of existing DCPs with creative, in-house solutions. For example, Molnar (2019) created a solution for real-time data collection to extend the function of a popular platform, and it required technical knowledge of at least two web development languages. Although available to the public, his solution was still produced with a limited scope in mind, and users may encounter additional constraints that could be challenging to navigate if they chose to implement it into their specific contexts. Part of the purpose of this contribution is to explore other DSR options with existing GAI integrations using R/Shiny.

R/Shiny for Data Collection

R is a programming language that has been around for many decades (R Foundation, n.d.). It excels at allowing users to work with data. The base R language has been extended to be more intuitive and generally easier to work with by Wickham (2019) and his architecture of the

tidyverse. The tidyverse is a collection of software packages (i.e., software add-ons) that work together well and have an underlying philosophy for how they operate. Shiny is an R package that lets you easily and intuitively build web applications with base R and tidyverse code. It handles technical complexities of things like client-server communication and front-end development with a layer of abstraction that makes application deployment much more feasible for researchers who may have limited experience in this domain.

With its relatively recent innovations introduced with the tidyverse, R has become more human-centered and focused on being useful to professionals who work with data (Wickham et al., 2019). The tidyverse is grounded in the actions commonly taken by professionals who import, clean, manipulate, visualize, model, and communicate within their data focused workflows. R is grounded in reliability while the tidyverse (which is fully written in R) values the interface between humans and computers, making this merger of technologies ideal in many respects. As R and the tidyverse are so useful for these types of data tasks, it is commonly used in higher education settings and some degree of expertise in R can often be found in departments of institutional research and effectiveness.

The Shiny web application framework extends the functionality of R and the tidyverse. Shiny provides a means to abstract the complexities of web development and create applications from R code. Data workflows and calculations are handled by R while the Shiny framework handles the server functions and user interface code for making the resulting applications interactive. For this reason, it can be ideal for collecting data and extending the limitations of a given platform. Kaufman (2020) recognized the limitations of existing DCPs and utilized Shiny to creatively explore what is possible for data collection. His research yielded web applications that implemented novel approaches with methods like randomized questions and randomized

intracoder reliability. One important finding was that the dynamic and flexible nature of R Shiny was ideal for executing complex calculations on data and that the code to implement these operations into web applications was mostly an extension of standard R usage with no additional programming languages needed.

Design Science Research

As noted, DCPs are robust but can be limiting (e.g., features locked behind paid tier levels, limited or no API integration, limited or no real-time data calculations) and present obstacles to HE data professionals who may desire to collect data in a specific way to best serve stakeholders. To investigate the use of GAI with R Shiny integrations into existing DCPs, a Design Science Research approach is adopted. DSR is rooted in solving problems and extending practically the abilities of the people who work in a given organization (Vom Brocke et al., 2020). The output of DSR is not only the artifact but is also knowledge gained from the process and result. Importantly, these resulting artifacts from the DSR process are infrequently the final system to be implemented. Instead, they are products that point to potential and possibilities bridging the gap between the human centric and the technological.

The central focus of DSR is the helpful artifact produced through a process of understanding the context, the problem, and the stakeholders (Peppers et al., 2007). The result can possibly yield research that is creatively unique, generalizable, and useful itself (Hevener et al., 2004). With these implications at the forefront of the illustrative demonstration offered in the paper at-hand, DSR is utilized in an example HE scenario to understand how it could be useful for producing context specific POC web application solutions to better serve stakeholders and go beyond the on-rails experience offered by existing DCP workflows.

Non-programmer Development

Software development can be characterized as performed best when executed in pursuit of solving a problem for stakeholders and when involving folks who have a very applied understanding of the issue to solve (Mehta et al., 2008). Typically, with modern development methodologies, this can involve several systems and professionals to derive an understanding from the group with knowledge of the issue, documenting the problem thoroughly, communicating it to the development professionals, evaluating the degree to which the developed solution solves the problem, and iterating on the application delivered (Sachdeva, 2016). When the non-programmer stakeholder who has a detailed understanding of the problem to be solved is empowered to create a POC solution, it is possible that a more useful solution could be developed faster and more efficiently. It is with this context in mind that the example SDR process is demonstrated in the research at-hand to illustrate how HE data professionals might create natural language coded POC web applications to support institutional goals.

Methodology

Conceptual Framework w. Justification

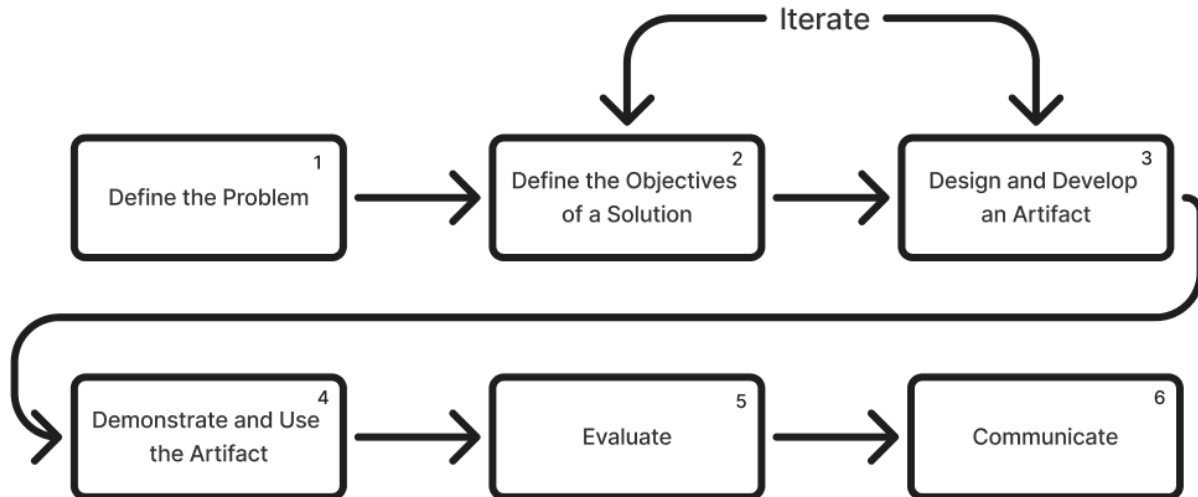
The purpose of this illustrative design science informed research is to demonstrate how a POC survey web application artifact could be created with conversational coding to explore the possibilities that exist for highly tailored and context driven HE data collection beyond the limitations of off-the-shelf DCPs. POC in this paper shares a similar definition used by Neto and authors (2018) where it is defined as a research activity that provides understanding, exploration, learning, and validation. The creation process is meant to be a practical one, usable by HE data professionals with limited coding experience. For this reason, we will use conversational coding with a GAI chatbot to fill in technical gaps and a hypothetical scenario to show how the DSR

framework could be used in the applied HE setting to yield a functional POC web application artifact.

Used in this DSR was an adapted version of the framework offered by Peffers and authors (2007) seen in Figure 1. The first step in the process is about understanding the unique problem and the value it has for stakeholders (Hevener et al., 2004). This grounds the entire DSR POC process and will largely determine the quality of the result. The next step is to define the objectives of the artifact and how they will relate to the problem. Contextual knowledge about existing solutions (if applicable) and how the proposed objectives improve these can be informative at this stage (Vom Brocke et al., 2020). The third step occurs iteratively between defining the objectives, and designing and developing the artifact where checks occur between what is made and what is intended. The illustrative artifact in this paper is the HE focused example POC web application. Next the artifact is demonstrated within the context where it is intended to be used and then evaluated to see if it is sufficient. This can lead to further development if necessary to refine the artifact to better support objectives. Lastly the results are communicated. In an applied setting, this communication could be directed to technical or funding stakeholders to be developed into a robust and complete computer application.

Figure 1

Adapted Design Science Research Framework



Note. This Design Science Research framework was adapted from Peffers et al., 2007.

Scenario

The DSR framework is presented in this paper as a tool that could be used in the applied setting to create a useful POC artifact. The work at-hand is a demonstration of the use of conversational coding situated within a hypothetical scenario to ground the DSR in an HE context.

Scenario Narrative

A professional at a university in the state of Tennessee works in the recruitment department that is responsible for increased enrollment. The professional believes that the university offers a unique value proposition compared to other postsecondary institutions in the state for salary earnings initially after graduation and debt incurred as a student. They have verified this by looking up related variables in the U.S. Department of Education's College Scorecard data (USCSD). The professional would like to i) present this USCSD dynamically to students where they could look up factual earnings and debt statistics for the universities and programs they are interested in, ii) display comparisons between their institutions and others in the state, iii) collect survey data to understand students perceptions of the results they found and

its influence, iv) and view the survey response data in a custom and helpful way on a dashboard. They have verified that these capabilities are beyond the scope of their current DCP by talking with their university data science team. The information technology department has indicated a willingness to work with the professional but a validated concept is needed before committing to development of a complete application.

Development Tools

The tools needed for this illustrative demonstration are the freely available R software and R Studio. R Studio is where the code will be entered and executed. A basic knowledge of R is helpful, but not strictly required. Also needed is access to Anthropic's Claude® GAI assistant which has free and very low-cost monthly subscription options and is accessible in most web browsers. Claude will serve as our GAI chatbot for conversational coding where we will enter text outlining our project and it will output code for us to paste to R Studio. The resulting application will be a file that can be sent to stakeholders directly and demonstrated from a local computer. One option for easy one-click deployment to the web is to use the hosting service at shinyapps.io which offers a free tier. Figures 2 and 3 show the interfaces for these applications. Also, a freely available application programming interface (API) key will be needed that can be retrieved from the USCSD website. Links to all these items where instructions and documentation can be examined are found in the appendix.

Figure 2

Basic Layout of the R Studio Application for Running R Code

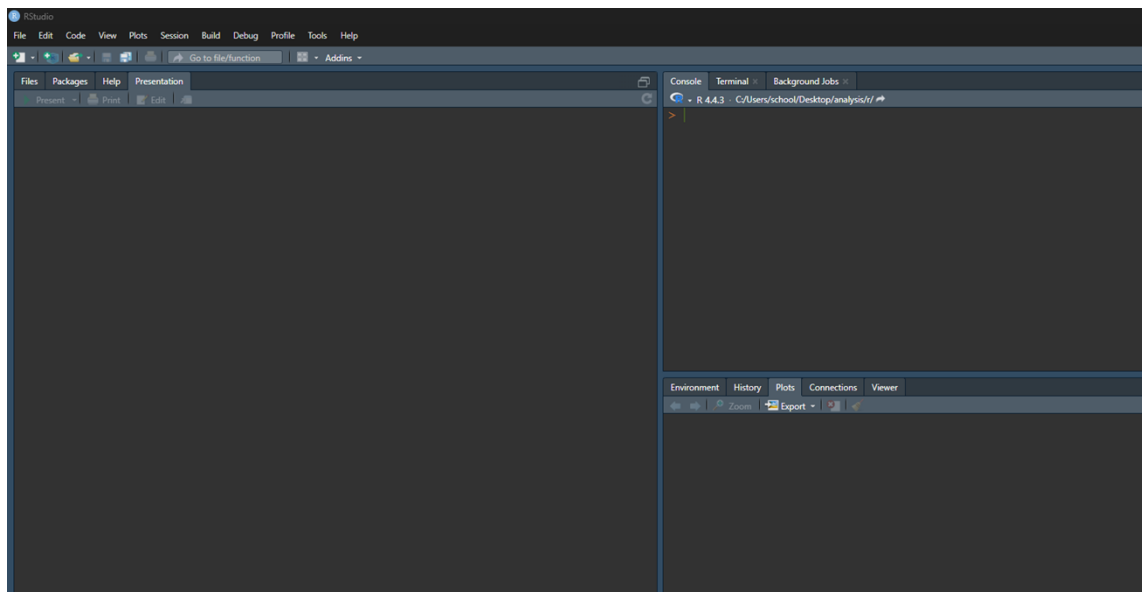
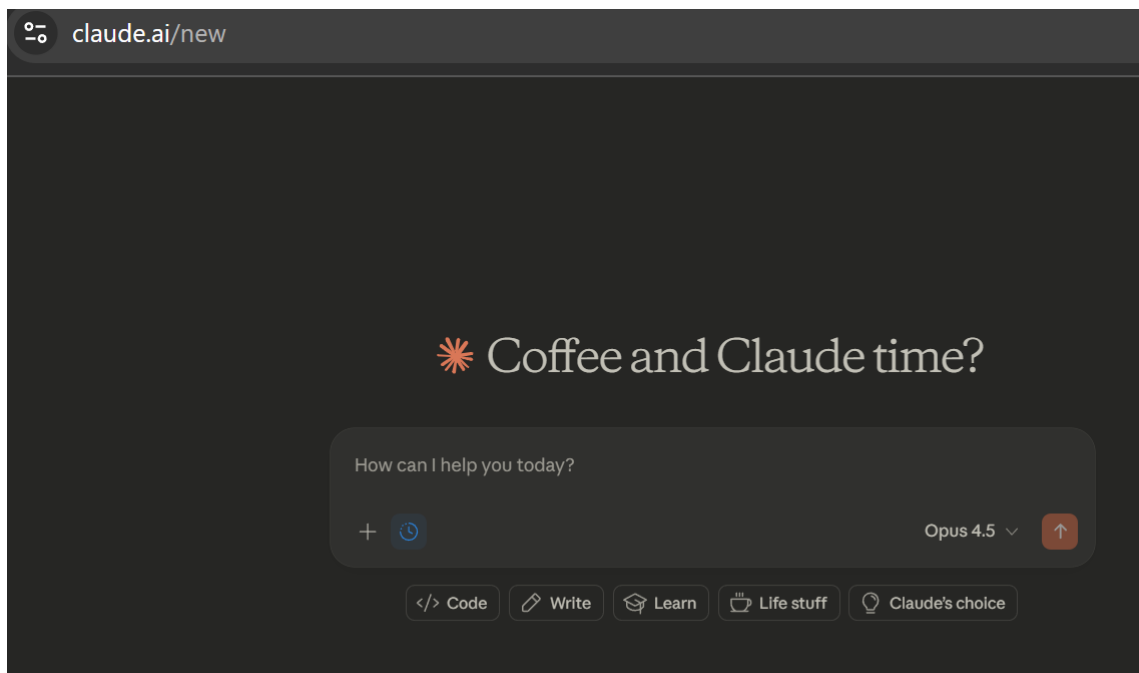


Figure 3

Basic Structure of Claude for Conversational Coding



Applied DSR Process

A hypothetical DSR process is presented here to arrive at the conversationally coded POC artifact. Each step is framed within the hypothetical scenario and offers a brief glimpse at

the types of understandings that could be sought with the understanding that the complexity of these could fluctuate based on the scope of the project.

Step 1: Define the Problem

A professional at a university would like to make the value proposition of studying at their institution more evident to potential students as they consider online the benefits and trade-offs of attending competing universities in terms of possible earnings and student debt incurred. The professional is focused on increasing enrollment numbers and understanding how the earnings and debt information learned by students about a given program and university influences students' college decisions. Also, they want to understand students' perceptions of potential earnings and student debt compared to actual values. The professional has noticed much conflicting data online about this topic and would like the information provided to be from the valid and reliable USCSD. Additionally, analysis of data via an export and import process can be cumbersome and they think that a flexible dashboard with calculations and metrics they define could help with this overhead.

Step 2: Define the Objectives of a Solution

The POC web application should allow students to choose from a defined list of universities and programs of study. The student should provide estimates for earnings initially after graduation and then some years after. Estimates should also be collected for the amount of debt the student expects to incur. Students see totals and comparisons for this data and then answer questions about their perceptions of it as well as submit an email address if they want university advisors to follow-up with them. The dashboard should show the results on a table along with helpful visualizations and a way to export and import data to analyze for testing the application and demonstrating to stakeholders with.

Step 3: Design and Develop an Artifact

The requirements were gathered and a prompt was constructed for the first iteration. Additionally, wireframes (i.e, a simple drawing of what you want the app to look like) could be constructed to guide this process. Although wireframing is beyond the scope of this present paper, the authors intend to create a paper to cover the topic in the future. Below we share the prompts that guided this process:

Page 1. *Hi Claude. I would like to create a student facing proof-of-concept application with R Shiny that utilizes the U.S. Department of Education College Scorecard Data API. Here is my API key that I think you will need to access that data: —¹. The first page of the app should present students with a list of universities to choose from that they may be interested in studying at in the state of Tennessee. Here is a list of the universities to include along with their Scorecard ID:—. Next ask the student about the instructional programs they are interested in. Here is a list of the programs we are interested in including along with their Classification for Instructional Programs (CIP) codes: —. Provide fields asking students to estimate what they expect to earn in their first year after graduation, their fourth year after graduation, and how much debt they expect to accumulate as students. Students should submit this data before being taken to the next page where comparison calculations are made with data fetched from the API. Use these three variables to make the calculations with: EARN_MDN_1YR, EARN_MDN_4YR, DEBT_ALL_STGP_EVAL_MDN. Create the first page and I would verify that it is correct before we proceed to the next page of the application.*

1

In compliance with federal regulations, personal identification and sensitive details have been omitted in this section.

Page 2. For the second page in the app, use the entered values from the student and the retrieved values from the API for the given institution and program of study. In one table present calculations to students showing what their expectations are for earnings and debt along with the actual median value and the difference in these two values expressed as a percentage. Show an additional table that shows a comparison of these values for the selected university and —. After viewing this data, students should be able to proceed to the third and final page in the app.

Page 3. The third page in the app should ask students about their agreement (i.e., 5 point Likert scale from Strongly Disagree to Strongly Agree) with the following statements: "This information was surprising to me."; "This information changes how I think about my plans." Include an option to enter an email address and indicate if they would like to be contacted by a university advisor to discuss university plans. Also include an open-ended response option for any other information the student may want to communicate and a button to submit the survey.

Dashboard. The dashboard in the POC should include a response log table that includes a generated ID, a time stamp for when the data were recorded, the institution of interest, the program of interest, the first-year gap percentage calculation, the Likert response for if the student was surprised, if they want to be contacted, and their email if they gave one. Score cards should be seen for the total count of responses, the average of the gap for expected and actual first-year earnings, the percentage of students who were surprised by what the data told them(i.e., agree or higher), and the percentage of students who want to be contacted. Include two data visualizations. One should be a bar chart that displays the total number of responses by the university of interest. The other should be a histogram showing the bins for the distribution of the gaps in perceived and actual first-year earnings.

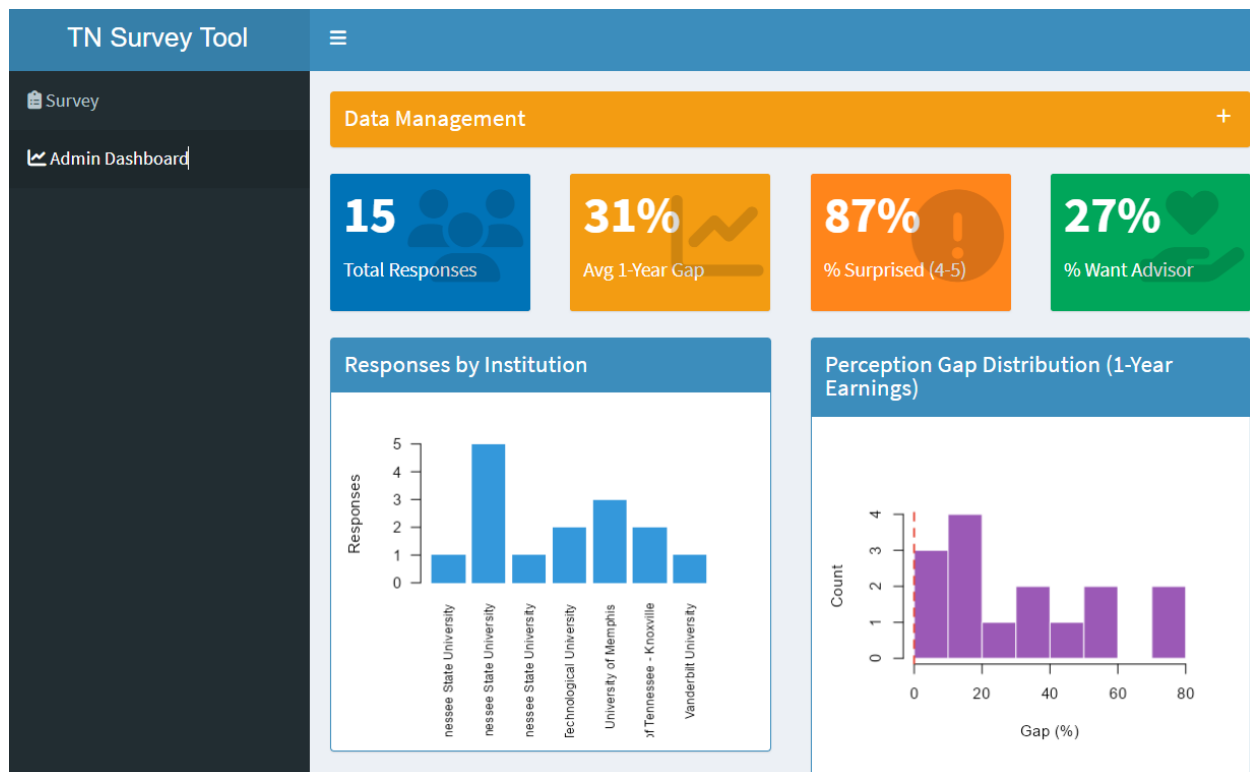
Step 4: Demonstrate the Artifact

The final POC web application is linked in the appendix and can be found online here:

https://sbndco-michael-littrell.shinyapps.io/University_Survey_App/. A partial image of the POC dashboard is seen in Figure 4.

Figure 4

Partial Image of the POC Dashboard



Steps 5 and 6: Evaluate and Communicate

As mentioned, when discussing the DSR framework, iteration occurs between the defining of the objective and the creation of the artifact. Additional iteration could also occur if deficiencies were discovered when demonstrating and evaluating (i.e., tested through use by stakeholders) the POC web application. In this example, the application would be tested to see if it aligned with the objectives and was useful to solving the initially identified problems to solve. If issues were identified, then additional prompts communicated to Claude could be made to

update the code. Iteration occurred in this example for building a quick way to manage the data on the dashboard where it could be easily loaded with data for testing and demonstration. The app was deployed from R Studio to shinyapps.io for easy hosting and demonstration. Once fully tested and the POC was shown to successfully align with the desired outcomes, communication with technology or funding stakeholders could commence for full development of the app.

Results

The purpose of this DSR-informed work is to explore data collection possibilities through the illustrative conversational coding development of a POC web application that exists outside the limitations of a DCP. We restate the research question:

Q1: In what ways can a proof-of-concept survey data collection application built with assistance from generative artificial intelligence extend the capabilities of institutional researchers beyond the limitations of existing data collection platforms?

As noted in this DSR study, the conversational approach to coding with Claude using R Shiny yielded a functional POC application that is obtainable by HE data professionals who may not have experience with software development.

The POC goes beyond what is possible and efficient for existing data collection platforms in several ways. The survey instrument itself is created as an informative tool where API calls collect up to date data from the USCSD to contextualize the responses of the student participants. Respondents are then able to see the data they entered about expectations of earnings and debt situated within actual median totals from the USCSD and helpful calculations made by the instrument are presented alongside data for a baseline comparison university. After viewing this information, the HE professional can collect survey data about how the student is influenced by it and how this understanding may shape the learner's college plans. A highly customizable

dashboard was created that allows for rapidly filling example observations for demonstration and for templated uploads.

A strength of using Claude® was that for any information not provided, it made reasonable guesses to fill in the gaps. As an example, it settled on using the shinydashboard package (i.e., a helpful app to extend the capabilities of base R) to create the dashboard in a simple and clean way even though this was not part of the original request. GAI is also helpful for understanding and learning code. Any section of the code that HE professionals may want to investigate in more depth and explore for customization can be dropped into chat with Claude® where a request for explanatory comments can be made.

Limitations

A limitation to this work was that it was applied to a hypothetical HE scenario and not one that exists in the real world. In addition to this, the authors have some knowledge of R and technical abilities that can influence their working methods even though every effort was made to rely only on natural language coding to produce the POC. While Claude® has given the HE data professional with limited coding experience much greater technical ability, practice and sustained effort for learning about conversational coding will improve the outcome of the results and the efficiency at which they are achieved.

Conclusion

Understanding technology and being able to leverage customizable adaptations to real-time student demands in data-driven decision-making processes is vital to HE institutions' ability to remain operationally efficient (Bollaert, 2025). In this sense, leveraging GAI to help HE professionals become capacity-builders for technological adaptability is essential (Tsai et al., 2019; Webber & Zhang, 2023). As demonstrated in this contribution, conversational coding of

POC web applications is not perfectly refined, but its current state points to much potential for applied use in HE settings for expanding the limitations of off-the-shelf DCPs, which are a central mechanism for data-driven decision-making processes. In one sense, having some control of the features of the software and how they operate offers the HE data professional with limited software coding skills a fresh perspective on what is possible and perhaps can compel them to consider survey mechanisms beyond what is offered from common platforms. Although successfully completing this type of data collection POC does require some learning for individuals who have not created computer applications before, the results are likely now more obtainable with assistance from GAI. Continued research in this area is therefore essential as conversational coding techniques are adapted or adopted at HE institutions as part of larger professional development or capacity building opportunities.

References

- Agasisti, T., & Bowers, A. J. (2017). Data analytics and decision-making in education: Towards the educational data scientist as a key actor in schools and higher education institutions. In Johnes, G., Johnes, J., Agasisti, T., López-Torres, L. (Eds.), *Handbook of Contemporary Education Economics* (p.184-210). Edward Elgar Publishing. ISBN: 978-1-78536-906-3 <http://www.e-elgar.com>
- Bollaert, L. (2025). Artificial intelligence: Objective or tool in the 21st-century higher education strategy and leadership? *Education Sciences*, 15(6), 774. <https://doi.org/10.3390/educsci15060774>
- Fosnacht, K., Sarraf, S., Howe, E., & Peck, L. K. (2017). How important are high response rates for college surveys?. *The Review of Higher Education*, 40(2), 245-265.
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS quarterly*, 75-105.
- Kaufman, A. R. (2020) Implementing novel, flexible, and powerful survey designs in R Shiny. *PLoS ONE* 15(4): e0232424. <https://doi.org/10.1371/journal.pone.0232424>
- Mehta, M., Anderson, D., & Raffo, D. (2008). Providing value to customers in software development through lean principles. *Software Process: Improvement and Practice*, 13(1), 101-109.
- Molnar, A. (2019). SMARTRIQS: A simple method allowing real-time respondent interaction in Qualtrics surveys. *Journal of Behavioral and Experimental Finance*, 22, 161-169.

- Neto, A. J. R., Borges, M. M., & Roque, L. (2018, October). A preliminary study of proof of concept practices and their connection with information systems and information science. *In Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 270-275).
- Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems*, 24(3), 45-77.
- R Foundation. (n.d.). *What is R?* <https://www.r-project.org/about.html>
- Sachdeva, S. (2016). Scrum methodology. *Int. J. Eng. Comput. Sci*, 5(16792), 16792-16800.
- Tsai, Y-S., Poquet, O., Gašević, D., Dawson, S., & Pardo, A. (2019). Complexity leadership in learning analytics: Drivers, challenges, and opportunities. *British Journal of Educational Technology*, 50(6), 2839–2854. <https://doi.org/10.1111/bjet.12846>
- Vom Brocke, J., Hevner, A., & Maedche, A. (2020). Introduction to design science research. In *Design science research. Cases* (pp. 1-13). Cham: Springer International Publishing.
- Webber, K. L., & Zheng, H. Y. (2023). *Big data on campus: Data analysis and decision making in higher education*. John Hopkins University Press.
- Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D. A., François, R., ... & Yutani, H. (2019). Welcome to the Tidyverse. *Journal of open source software*, 4(43), 1686.

Appendix

Claude - <https://claude.ai/>

College Scorecard API Key - <https://collegescorecard.ed.gov/data/api-documentation/>

POC Application Artifact - https://sbndco-michael-littrell.shinyapps.io/University_Survey_App/

R Software - <https://cran.rstudio.com/>

R Studio - <https://posit.co/download/rstudio-desktop/>

Shinyapps.io - <https://www.shinyapps.io/>