

Computing with CORGIS: Diverse, Real-world Datasets for Introductory Computing

ABSTRACT

Bringing introductory computing to non-CS majors means creating a curriculum that will appeal to students from diverse disciplines. Many educational theories emphasize the need for introductory contexts that align with students' long-term goals and are perceived as useful. Data Science, using algorithms to manipulate real-world data and interpreting the results, has emerged as a field with cross-disciplinary value, and has strong potential as an appealing context for introductory computing courses. However, it is not easy to find, clean, and integrate datasets that will satisfy a broad variety of learners. The CORGIS project makes it easier for instructors to incorporate data science into their classroom. We provide over 50 datasets, from history, politics, medicine, education, and more. The CORGIS infrastructure makes it easy to integrate a new dataset with simple libraries for Java, Python, and Racket, so that introductory students can write programs that manipulate real data. Web-based tools also allow learners to visualize and explore datasets without programming, enabling data science lessons on day one. We have incorporated CORGIS assignments into an introductory course for non-majors to study their impact on learners' motivation, with positive initial results. We are hopeful that external adopters will find our tools and materials useful.

CCS Concepts

•**Information systems** → **Information integration**; Data management systems; •**Social and professional topics** → *Computational thinking*; *Model curricula*; Computing literacy;

Keywords

Computational thinking; data science; pedagogy; motivation; authenticity; real-world data; big data

1. INTRODUCTION

As computing skills become required for an increasing number of disciplines, CS educators are meeting the demand by creating courses for non-CS majors. Students are drawn from many disciplines, including the sciences, arts, and humanities, and they have myriad, divergent career paths before them distinct from those of Computer Science majors. This different population needs a different approach, one that all students will perceive as authentic and beneficial, while simultaneously enabling problems of a computational nature [9].

We suggest that most students would benefit from computational techniques that empower them to manage the ever-growing amounts of data present in every field [14]. Therefore, a Data Science context, where students write algorithms to manipulate real-world data and interpret the results, could provide a compelling context. However, integrating data into introductory courses creates challenges for instructors, both pedagogically and technologically. Finding many, varied datasets can be difficult, and they often require cleaning to be suitable for beginners (e.g., to remove missing values, to scale it to the appropriate size, to choose a convenient format). To overcome these problems, we present here the open-source CORGIS project, the **C**ollection **O**f **R**eally **G**reat and **I**nteresting **d**ata**S**ets, which makes a wide variety of student-ready datasets available. Our materials are free and open-source, available at <http://tinyurl.com/sigcse-corgis>.

This paper begins with a brief review of relevant educational theory and existing projects in this space. We then briefly describe the technical innovations of the CORGIS project and its pedagogical affordances, and specific ways that it can be used in a course. We report and evaluate results of an intervention conducted using the software. Finally, we discuss the future of the project.

2. EDUCATIONAL THEORY

Educational Theory grounds the development of the CORGIS project. We use Situated Learning Theory [13] to better understand how students learn, and the MUSIC Model of Academic Motivation [12] to better understand why students choose to engage.

Situated Learning Theory suggests that authentic contexts are crucial for learners. Originally proposed by Lave and Wenger, SL Theory argues that tasks in the learning environment should parallel real-world tasks, in order to maximize the authenticity of the experience [13]. Some interpretations of this theory draw a distinction between the content

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGCSE '17 Seattle, Washington USA

© 2016 ACM. ISBN ...\$15.00

DOI:

(e.g., learning to program) and the context (e.g., by creating a video game), and stress that proper contextualization is important for students’ comprehension and investment [5].

In our research, we rely on the MUSIC Model of Academic Motivation [12]. Built as a meta-model, it incorporates many existing theories and is tailored particularly for education. The model distinguishes between students’ situational *interest* in an academic activity and their sense of the *usefulness* of the activity to their long-term career goals. We connect this distinction to the different contexts available to introductory courses. Creating games and animations, for example, appeals to students’ situational interest, while we posit that data science will appeal more to students’ sense of usefulness. The MUSIC model also incorporates students’ expectation for appropriate *Successes* when learning, their sense of *eMpowerment* within the activity, and their perception of their instructors’ *Caring* towards them.

From these two theories, we hypothesize that Data Science will be a beneficial context for students, since students will perceive the context as closely related to their career goals.

3. RELATED AND PRIOR WORK

The use of data analysis for contextualization represents an actively growing movement [1, 16, 10, 6]. Upper division courses have employed situated learning experiences using data of varying size and complexity for several years [7, 17]. However, this prior work has little evaluation of the impacts of data science in introductory education [16, 6].

The RealTimeWeb project (RTW) [2] made it easy for instructors to bring web-based, rapidly-updating data into their introductory classrooms by writing light-weight specifications of remote data streams. CORGIS is similar to RTW, moving from a small collection of web-based data to a wide collection of local data. The change was motivated by the authors perceived scarcity of quality, real-time datasets. And, while the technical challenges of integrating real-time data are interesting, they are of secondary importance to the value of having a diversity of relevant datasets.

The BigDataCSE project of Hamid [11] takes real-time data access in a different direction by reducing the technical requirements on instructors even further. The library uses sophisticated reflection techniques to automatically infer the structure of data, so that students can access any desired URL endpoint and receive structured data. Although it provides a flexible architecture, this project does not attempt to provide datasets, making it more appropriate for advanced students who can find their own data to work with.

The BRIDGES project provides students with visualizations of their algorithms on datasets [4]. BRIDGES does not focus on organizing datasets directly, but incorporates existing datasets (including some directly from the CORGIS project). The use of visualizations can help students understand the interaction between their algorithms and data.

4. DESIGN

The CORGIS project aspires to have multiple (3-4) datasets relevant to each major career path that potential students might seek. To achieve this, we draw on data from wide-ranging, open-access sources including governments, research publications, journalists, non-profits, and industry. Figure 1 provides an overview of how datasets are added to the CORGIS project.

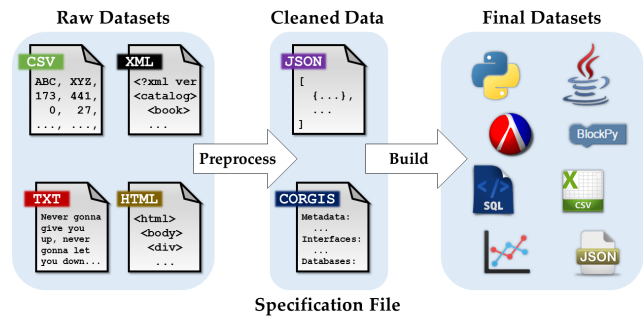


Figure 1: How datasets are added to CORGIS

First, real-world data is collected and preprocessed into a “clean” state. Most of the human effort for the project comes at this phase, as organizing a dataset requires expertise. Figure 5 shows a representation for the cleaned structure of a dataset. Every dataset becomes a list of hierarchical maps (a tree), where the leaf nodes are simple data types: numbers, text, and booleans.

Next, high-level specification files are written to describe the metadata, fields, and interfaces for a cleaned dataset. All CORGIS datasets have interfaces that can be used to slice the data. The weather dataset, for example, provides “Get Temperature” and “Get Past Temperatures” that return an integer and a list of integers, respectively. These specification files can be validated by the system using a custom compiler.

The CORGIS infrastructure uses the specification file and associated dataset to automatically generate a number of student-ready materials.

Language Libraries: The CORGIS project currently supports the generation of Python, Java, and Racket libraries. Figure 2 gives examples of how a generated library can be used in these languages. Code and supporting documentation for a library are generated by filling out templates using the Jinja2 templating library. Interfaces from the specification file generate functions that execute queries. A SQLite database is also generated to store the data, rather than the original JSON file, for performance reasons.

```

# Python
import crime
crimes = crime.get_all()

; Racket
(require crime)
(define reports (crime-get-all))

// Java
import org.corgis.crime.StateCrimeLibrary;
import org.corgis.crime.domain.Report;

public class Main {
    public static void main(String[] args) {
        StateCrimeLibrary stateCrimeLibrary =
            new StateCrimeLibrary();
        ArrayList<Report> reports =
            stateCrimeLibrary.getAllCrimes();
    }
}
  
```

Figure 2: Example of using a CORGIS library

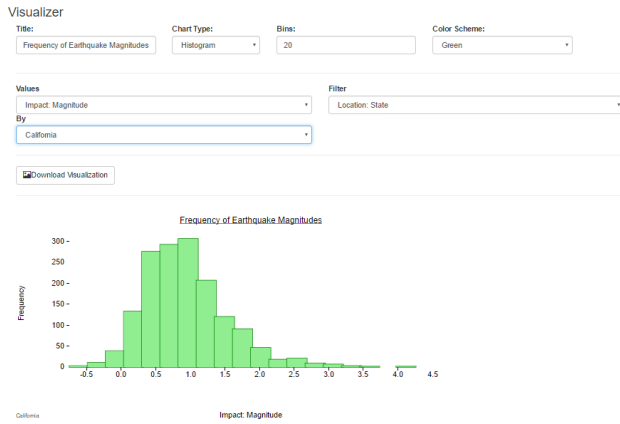


Figure 3: The CORGIS visualizer

Visualizer The Visualizer allows students to engage with datasets in their browser without programming, as shown in Figure 3. Users can generate histograms, line plots, scatter plots, and bar charts for any supported dataset. Bar charts are categorized by the indexes from the specification file. Datasets can also be filtered on indexed fields when generating charts. Although not as powerful as a full programming environment, the Visualizer empowers students to work with real-world data without specialized training.

Explorer Separate from data content, students can explore the structure of a dataset using the Explorer. The hierarchy of data is explored through chained windows. Clicking a field in a map, for instance, might open a new map’s window, which in turn might have atomic fields or links to further maps.

Raw Data Files Language bindings simplify access to datasets, but some instructors may want students to use more traditional methods. Therefore, we expose datasets in popular raw formats: JSON, SQL, and CSV. The SQL database represents each child map as a distinct table, linked with a primary key. The CSV table is a flattened representation of the data, using the key hierarchy to disambiguate column names.

The CORGIS Gallery Language bindings, data files, and the Visualizer are all accessible through a user-friendly Gallery, shown in Figure 4. Every dataset is documented with a list of content tags to help students search for a relevant dataset.

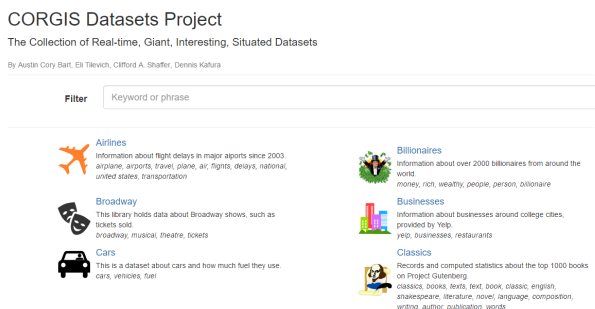


Figure 4: The CORGIS gallery

5. ASSIGNMENTS

CORGIS can be integrated into introductory courses in a number of ways. In this section, we describe how we have used the tools in an existing course on Computational Thinking for non-majors. This course teaches basic programming in a Data Science context. Much of the CORGIS development has been driven by the needs of the course.

The major motivational goal of integrating CORGIS is to align assignments with students’ career goals. However, assignments also benefit from the wide variety of the CORGIS collection. Learners can choose their own datasets to explore things related to their own interests and career goals, increasing their sense of agency.

5.1 Exploratory Analysis

The Visualizer facilitates early exploration of datasets, without the need to program. In the beginning of our course, students generate graphs in groups (each group is assigned a dataset) and then on their own (free to choose their own dataset). They are tasked with creating visualizations to explore questions about the distribution, trends, and relationships of the data. For example, in the crime dataset, they can identify the downward trend of violent crime rates over time in different states. This is an opportunity to discuss how complex, real-world entities and phenomenon can be represented with computable abstractions (e.g., numbers). Additionally, this gives students practice with selecting and interpreting different kinds of charts. In our experience, many students struggle with aspects of graphs such as distinguishing bar charts and histograms, or knowing when to use line plots.

5.2 Practice Problems

When students start programming, we give them practice problems contextualized with CORGIS datasets. Although the scope of these problems is similar to those found in systems like CodingBat [15], the problems can be more realistic. The complexity of using complete datasets is avoided by using the simpler interfaces exposed for libraries. For example, students might be tasked with writing a program to print whether an umbrella is necessary depending on the weather in their current city (requiring only a function call to the Weather library, an if-statement, and the print statement). A wide range of programming topics can be contextualized with the CORGIS libraries, including collection-based iteration, decisions, printing and visualization, and functions. In our course, datasets are incorporated through a block-based programming environment [3] and then in a regular Python programming environment, but the libraries and raw datasets could be used in a variety of development environments.

5.3 Data Mapping

A learning goal in our course is for students to be able to navigate and manipulate complex data structures (e.g., nested lists and dictionaries). We direct students to create a hand-written “data map”, like the one in Figure 5. This diagram visually describes the structure of the dataset, and students use it as a guide to developing the necessary code constructs (e.g., iteration, dictionary access), much as they would navigate a real map to find a path in a maze. Because of the vastness of most datasets’ structure, students must be judicious in what branches and leaves of the structure they

were male and half were female. Students came largely from the arts and humanities, with few students from the sciences. The distribution of years was skewed heavily towards Sophomores and Juniors, with only a third of the class being Freshman and Seniors.

Students were asked to rate their agreement with 26 statements on a 7-point likert scale (from “Strongly Disagree” to “Strongly Agree”). The first 25 statements were the cross-product of two sets of five aspects. First were the main course components: learning about abstraction, writing programs, real-world data, social ethics of computing, and working in small groups (cohorts). The second set were the elements of the MUSIC model: their belief in whether they had a choice, their interest, their sense of the usefulness, their sense of success, and their belief that the instructors cared. So an example statement would be “I believe that it was useful to my long-term career goals to learn to write computer programs”. The last statement, unrelated to the others, was their intent to continue learning computing, either informally (e.g., an online course) or formally (e.g., another Computer Science course).

Figure 8 presents raw results from the survey. Overall, most of their responses were encouraging, showing mostly positive answers (no student marked “Strongly Disagree” for any statement). We found that their interest in learning course components generally outweighed their sense of the usefulness to learning with respect to their long-term goals. They indicated that working with data related to their own major to be more useful for their career goals than learning to write programs. Students felt empowered and cared for by their instructors. They also indicated that they felt successful in learning the material. The lukewarm response to the last statement, students’ intent to continue, is expected since it is not the goal of the course to recruit new students into computing. However, we found it disappointing that not a single student marked “Strongly agree” for that element.

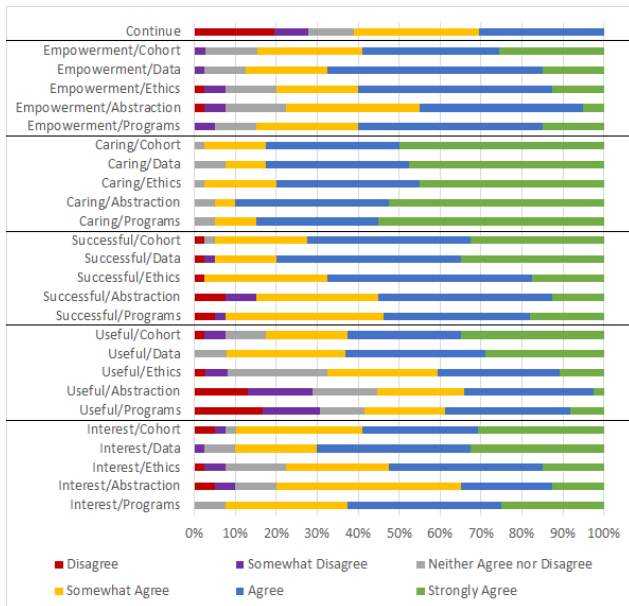


Figure 8: Survey results across motivational and course components

	M	U	S	I	C
Abstraction	45.8**	69.9**	61.4**	48.8**	19.2
Cohort	9.0	11.3	3.6	7.5	7.9
Data	13.8	30.9	29.4	29.7	21.9
Ethics	30.8	48.5**	41.8**	32.3*	18.6
Programs	43.7**	82.3**	60.0**	63.8**	7.3

Figure 9: Correlation between students’ intent to continue vs. components of the course with respect to motivational components (at end of semester)

Table 9 reveals a potentially important connection between students’ motivation with respect to the content and context as compared to their long-term intent to continue in computing. The vertical categories represent components of the course, while the horizontal categories represent elements of the MUSIC model. So, for example, the intersection of the row “Data” and column “I” should be interpreted as “The correlation between students’ interest in learning about real-world data related to their major and their intent to continue learning computing”. Bolded items are significantly correlated. Although the correlations are modest, there is a strong connection between students’ sense of the usefulness of learning the core course content (and, to a lesser extent, their self-efficacy and interest towards the content) and their intent to continue. However, there is no significant correlation between students’ sense of the usefulness of learning to work with real-world data (the course context) and their intent to continue. Note that this is despite students expressing more positive responses towards the usefulness of working with real-world data vs. programming. This makes sense, since students will perceive continuing in Computer Science as being about programming, not about data science. Our preliminary interpretation of this result is that, if we wish to encourage students to continue learning computing, we should convince them of ways that learning to write programs and learning computing can directly benefit their careers. Although the small sample size does mitigate the finding, and there are a number of other possible interpretations, it does suggest a worthwhile avenue for future research.

7. CONTRIBUTION

In this paper, we have described a number of innovations that we think will benefit the introductory computing education community. At its core, the CORGIS project represents a concerted effort to make many excellent datasets available for introductory computing students. We have introduced a novel architecture to support this process, making it easy to convert an existing dataset into a wide variety of student-ready materials. Over 50 datasets have already been incorporated into our collection, spanning across disciplines such as education, health, history, and politics. Not only do we provide beginner programmer-friendly libraries in multiple common introductory languages, we also provide web-based tools for quickly accessing the datasets without programming. We describe a number of ways these libraries and tools can be used in a course in order to satisfy both conceptual and programming learning objectives. Finally, we report on survey results of the integration of these datasets into a computational thinking course. These results reveal not just the success of the integration, but also suggest a

potential path for bringing students more solidly into computing.

8. CONCLUSIONS AND FUTURE WORK

The RealTimeWeb and CORGIS projects have been developed over the past 4 years. At this point, it is useful to reflect on the future work needed for the project.

First, further research must be done to evaluate the impact of this type of context on learners' cognition and motivation, similar to the longitudinal studies performed by the Media Computation project [8]. Second, the quantity, quality, and diversity of datasets needs to continue to grow in order to meet the demand of students entering introductory computing. Although understanding the impact on learners is relatively straightforward, expanding the project's collection raises a number of technical issues.

A major limitation of dataset preparation is the expertise required to convert datasets into a format suitable for processing. Although there is significant dedicated research to lowering barriers for this process, it is still the work of a seasoned programmer. Further, every dataset demands some level of domain knowledge to correctly and meaningfully arrange the fields. We currently use only open-access, unprotected datasets, thus limiting the field sharply. One unexplored direction is the creation of artificial datasets based on known characteristics of existing but unsuitable datasets. For example, a detailed dataset representing hospital records would normally be protected by HIPAA. But an artificial collection of records might be generated from the real data, suitable for open use. Although such datasets would not allow students to discover trends and facts about reality, we could use statistical techniques to ensure some relative accuracy of the data. Further research must be conducted to see if this is technically feasible and still perceived as authentic by students.

We hope that interested readers will try out the materials at <http://tinyurl.com/sigcse-corgis>, and consider submitting their own datasets to expand our collection.

9. ACKNOWLEDGMENTS

We gratefully acknowledge the support of the National Science Foundation.

10. REFERENCES

- [1] R. E. Anderson, M. D. Ernst, R. Ordóñez, P. Pham, and S. A. Wolfman. Introductory programming meets the real world: Using real problems and data in CS1. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, SIGCSE '14, pages 465–466, 2014.
- [2] A. C. Bart, E. Tilevich, S. Hall, T. Allevato, and C. A. Shaffer. Transforming introductory computer science projects via real-time web data. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, SIGCSE '14, pages 289–294, 2014.
- [3] A. C. Bart, E. Tilevich, C. A. Shaffer, and D. Kafura. Position paper: From interest to usefulness with BlockPy, a block-based, educational environment. In *Blocks and Beyond Workshop (Blocks and Beyond)*, 2015 IEEE, pages 87–89, 2015.
- [4] D. Burlinson, M. Mehedint, C. Grafer, K. Subramanian, J. Payton, P. Goolkasian, M. Youngblood, and R. Kosara. BRIDGES: A system to enable creation of engaging data structures assignments with real-world data and visualizations. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, pages 18–23. ACM, 2016.
- [5] J.-I. Choi and M. Hannafin. Situated cognition and learning environments: Roles, structures, and implications for design. *Educational Technology Research and Development*, 43(2):53–69, 1995.
- [6] P. DePasquale. Exploiting on-line data sources as the basis of programming projects. In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '06, pages 283–287, 2006.
- [7] A. E. Egger. Engaging students in earthquakes via real-time data and decisions. *Science*, 336(6089):1654–1655, 2012.
- [8] M. Guzdial. Exploring hypotheses about media computation. In *Proceedings of the ninth annual international ACM conference on International computing education research*, pages 19–26. ACM, 2013.
- [9] M. Guzdial and A. E. Tew. Imagineering inauthentic legitimate peripheral participation: an instructional design approach for motivating computing education. In *Proceedings of the second international workshop on Computing education research*, pages 51–58. ACM, 2006.
- [10] O. A. Hall-Holt and K. R. Sanft. Statistics-infused introduction to computer science. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, SIGCSE '15, pages 138–143, 2015.
- [11] N. A. Hamid. A generic framework for engaging online data sources in introductory programming courses. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, 2016.
- [12] B. D. Jones. Motivating students to engage in learning: The MUSIC model of academic motivation. *International Journal of Teaching and Learning in Higher Education*, 21(2):272–285, 2009.
- [13] J. Lave and E. Wenger. *Situated learning: Legitimate peripheral participation*. Cambridge university press, 1991.
- [14] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers. Big data: The next frontier for innovation, competition, and productivity. 2011.
- [15] N. Parlante. Codingbat. *Com* (Retrieved 1/08/2011 from <http://codingbat.com>, 2011), 2015.
- [16] D. G. Sullivan. A data-centric introduction to computer science for non-majors. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, SIGCSE '13, pages 71–76, 2013.
- [17] L. Torgo. *Data Mining with R, learning with case studies*. Chapman and Hall/CRC, 2010.