Received: 20 April 2020 | Revised: 21 October 2020 | Accepted: 21 October 2020

DOI: 10.1111/test.12243

ORIGINAL ARTICLE

WILEY

Introductory data science across disciplines, using Python, case studies, and industry consulting projects

Jana Lasser[1,2,3] | **Debsankha Manik**[2] | **Alexander Silbersdorff**[3,4] | **Benjamin Säfken**[3,4] | **Thomas Kneib**[3,4]

[1]Complexity Science Hub Vienna, Vienna, Austria

[2]Max Planck Institute for Dynamics and Self-Organization, Göttingen, Germany

[3]Centre for Statistics, Georg August University Göttingen, Göttingen, Germany

[4]Campus-Institut Data Science, Georg August University Göttingen, Göttingen, Germany

Correspondence
Jana Lasser, Complexity Science Hub Vienna, Josefstädterstrasse 39, 1080 Vienna, Austria.
Email: lasser@csh.ac.at

Funding information
Stifterverband; WOA Institution: GEORG-AUGUST-UNIVERSITAET GOTTINGEN Blended DEAL: ProjektDEAL

Abstract

Data and its applications are increasingly ubiquitous in the rapidly digitizing world and consequently, students across different disciplines face increasing demand to develop skills to answer both academia's and businesses' increasing need to collect, manage, evaluate, apply and extract knowledge from data and critically reflect upon the derived insights. On the basis of recent experiences at the University of Ttingen, Germany, we present a new approach to teach the relevant data science skills as an introductory service course at the university or advanced college level. We describe the outline of a complete course that relies on case studies and project work built around contemporary data sets, including openly available online teaching resources.

KEYWORDS

teaching, data literacy, data science, interdisciplinary, jupyter notebook, python

1 | INTRODUCTION

Data science and its applications are increasingly ubiquitous in the rapidly digitizing world and consequently students across different disciplines face increasing demand to develop skills and awareness [27,30] to answer needs across all sectors to collect, manage, evaluate, apply and extract knowledge from data and critically reflect upon the derived insights. Against this backdrop, competencies to implement essential data analysis independently and to develop a basic understanding of more advanced processes and procedures used by data scientists in order to collaborate with them in a specific field of work are deemed desirable if not outright necessary in the future.

As part of a joint initiative of several German universities and German businesses, the authors of this article have developed a "service-course" that aims to teach fundamental data competencies to students from all disciplines at the University of Göttingen. See https://www.stifterverband.org/data-literacy-education for further information on this collaboration. The course especially addresses those students from outside STEM-subjects (science, technology, engineering, and mathematics) who generally have no prior experience with statistics or programming from their school education and highlights the importance of data competencies in prospective occupational fields for those students at the outset of their studies.

We aim to provide all participating students with a fundamental understanding of the concepts and procedure of data science and motivate a fair share of them to pursue further courses geared to evolve their competencies in that regard. Moreover, the course aims to convey

not only competencies from the domains of statistics and computer science but equally aims to develop the soft skills associated with data analysis, such as communicating the results in the context of tasks both inside and outside of university. The course builds on some role models, most prominently among them data8 (see http://data8.org/), the data science course developed at the University of Berkeley. Other similar courses are presented in [3,13]. However a number of factors and components of this course bring together both well-established and current pedagogies and practices of interest and value for current and future learning in introductory data science across disciplines.

Despite a general agreement regarding the importance of statistics and programming competencies in most ministries, the complex federal structure of school education in Germany and the general problems of overhauling established school curricula yield the given status quo with no systematic programming, data science or statistics training prior to university for most students. Hence this course cannot assume even the basic statistics background commonly found in at least middle school in so many countries today. In addition, students choosing to do the course are not only in areas which are traditional foci of non-STEM statistics courses (such as Business), but also in Humanities programs, including Linguistics, Archaeology and History. Such diversity of interests is catered for in split tutorials, which also use contemporary, complex and non-traditional data sets. What computer programming and how much are significant considerations in designing introductory data science courses. Assuming no prior programming experience left us free to choose a programming language. We chose to use Python despite the increasing use of R and R-based products in statistics courses and new data science courses.

In the final phase of the course, a mandatory project is carried out in authentic workplace-linked data investigations involving all aspects of data science at an introductory level. The importance of authentic experiential learning of the full statistical data investigation process has long been recognized, but facilitating this at the introductory level has proved challenging. Data science adds more challenges to this through requirements in data wrangling, cleaning and visualization. Therefore the course also builds on and extends to contemporary needs in learning from data, the objectives of courses with a particular focus on first hand real investigative experiences, such as [2,4,5,16,32,35].

In this article, we describe some details of the course, including the learning progression phases, tutorial work, projects, and openly available online teaching resources, consisting of slides, videos, exercises and solutions. Because the course is so new, only some initial evaluations are available; these are included in the discussion.

2 | COURSE OUTLINE

Within the teaching landscape of our university, this course is intended as a *service course* to introduce students from all disciplines to data science. The structure of the course is geared towards motivating students from the outset while gently introducing them to important core skills required for working with data or cooperating with data scientists. In addition, the interdisciplinary nature of the course and the time constraints imposed by the students' curricula require an approach that evades any discipline-specific overload, while also giving the students from the participating disciplines first insights into the nature of the applications of the competencies conveyed to them in the course.

Regarding the required core competencies for all these students from different disciplines, the target was to develop basic competencies allowing the students to independently find, read and clean data for further analysis. Considerable time was then devoted to developing the students ability regarding explorative and visual data analysis. Finally, we aimed for a fundamental understanding regarding inference and prediction on the basis of statistical models. While all of these desired processes can be taught by (often field specific) point-and-click interfaces - and previously mostly were in most non-STEM disciplines, this course specifically opted to pursue this introductory course on the basis of a programming language, following other recent initiatives to clear this methodological hurdle at the outset of the statistical education [6,34]).

The key argument for endeavoring to directly start with a programming language (and thereby against using spreadsheet programmes like Excel or OpenOfficeCalc and mostly interface-based data analytics programs such as SPSS or Gretl) were, that by using a programming language students were directly nudged towards thinking algorithmically [8] rather than looking for one single comprehensive process, thereby training a fundamental skill of analytical processes at large. In addition, the use of the programming language has the obvious advantage that students are introduced to the concept of programming. This is paramount for subsequent more advanced courses which will hopefully be attended by a substantial share of the courses' participants.

Concerning the programming language, we opted for Python. There are several reasons for choosing Python over other programming languages such as R, Java, SAS, STATA, or MATLAB: Python is a syntactically simple programming language, which facilitates the learning of basic programming concepts [1,14,19]. Additionally, Python is open source, eliminating the need for the acquisition of costly licenses. Python is highly prevalent in industry [24,31] and has a rich and thriving ecosystem of libraries for scientific computing [28], enabling

students to directly translate their competencies to potential applications in their later jobs in industry or research. Lastly, Python is a general purpose programming language, allowing for easy extension of the learned skills into different application areas such as image analysis, natural language processing, data mining or machine learning in subsequent lectures that build on our introductory course.

The course consists of a weekly lecture for all students. The lecture is accompanied by weekly tutorials in which students are split into groups with each group working on one of currently four domain-specific case studies (see Appendix A.1). Content-wise, the course is split into three phases of about equal length and importance: (I) teaching of basic programming skills in Python (II) teaching of data analysis methods and application in a case study (III) outlook and work on a project, which is used for student assessment. The time line of the lecture and accompanying tutorial on a week-to-week basis are shown in Table 1. The skills taught throughout the three course phases can be mapped to the skills making up the definition of "data acumen" [27], which we extend by a dedicated sixth skill for "data visualization":

- Combine many existing programs or codes into a "workflow" that will accomplish some important task;
- "Ingest," "clean," and then "wrangle" data into reliable and useful forms;
- Visualize data;
- Think about how a data processing workflow might be affected by data issues;
- Question the formulation and establishment of sound analytical methods; and
- Communicate effectively about properties of computer codes, task workflows, databases, and data issues.

Here, the lecture focuses predominantly on skills (d) and (e), whereas the tutorials focus on skills (a) and (b). Visualization (c) is used and taught continuously in both the lecture and tutorial to illustrate the application of other skills and motivate students. Skill (f) is predominantly acquired by the students during their work on projects and subsequent presentation of results. We note that we cover skill area (a) by introducing the students to a programming language (Python), which - together with existing third-party libraries - incorporates all necessary functionality to establish a data science workflow. In Table 1, mappings of lecture and tutorial content to these six skill areas are indicated by their respective letters from the list above.

The lecture is conceptualized as a blended learning exercise entailing classical input by a lecturer as well as live coding sessions and on-demand videos for key concepts that students can watch at home. In phases (I) and

TABLE 1 Time line of the semester-long lecture and accompanying tutorials

Phase	Week	Lecture	Tutorials
I	0	motivation and organization	
	1	causality, correlation[e]	tech-check
	2	Python, Jupyter Notebooks[a]	using Python as a calculator[a]
	3	data types[a,b]	text, numbers, lists, tables[a,b]
	4	tables, incomplete data[c]	logic, conditionals[b]
	5	mean, median[e]	functions, data types[b]
II	6	histograms, frequency tables[c]	data acquisition & cleaning[b]
	7	scatter plots, time series[c]	histogram, descriptive stats[c]
	8	regression[e]	scatterplot[c]
	9	clustering algorithms[e]	time series[c]
	10	bias[e]	regression[a,b,e]
	11	project selection	inference, bias[a,b,e]
III	12	open topic: data ethics[e]	work on projects[f]
	13	open topic: data protection[d,e]	work on projects[f]
	14	open topic: machine learning[e]	work on projects[f]
		project presentations[f]	

Note: Skills covered in the respective lecture and tutorial units corresponding to the "data acumen" definition [27] plus "data visualization" are indicated as superscribed letters (see list above). Note that the project presentations take place during the semester break, several weeks after the end of the lecture, as students require additional time to work on their projects between the last lecture and the project presentations.

(II), the content of the lecture serves two goals: (a) to quickly introduce students to the programming and statistics concepts necessary to work on the case study in the tutorials and (b) to present motivating examples of data science applications. In phase (III), the lecture content is designed to address current topics in data science and provide an outlook to domain specific applications that serve as a connection to further lectures that build on this introductory course.

During the tutorials, the concepts introduced in the lecture are applied to domain-specific case studies constructed around current and domain specific data sets. Every case study is designed as a series of exercises and solutions that build on each other and iterate the general phases of data acquisition, data cleaning, data exploration and visualization and the subsequent answering of research questions using inference. All case studies containing the individual tutorial exercises and solutions are freely available [23]. Learning in the tutorials follows a student-centered approach: Students are actively encouraged to look for solutions to coding problems on their own and help each other before they approach tutors with questions. Tutors are instructed to coach students on how to interpret error messages and use online resources such as library documentations or Stack Overflow (https://stackoverflow.com/) to solve problems they encounter while working on the exercises. This is deliberately done to mimic the iterative process of failing and learning from failure which is inherent to programming. Consequently, students develop communication strategies and problem solving skills by learning together [11].

To accommodate the different skill levels of the heterogeneous students, exercises are divided into optional and non-optional parts. Non-optional exercises are designed in a way that any student-including those with no prior knowledge-should be able to complete them during the course of the weekly, two-hour tutorial. Optional exercises are more challenging or provide additional domain-specific context and are aimed at more apt students and those interested in a deeper understanding.

The course assessment is specifically designed to convey substantial practical experiences to the students, which is known to be in high demand for their later vocational development. The assessment consists of work on a project in small teams of 2 to 3 students each. Projects are designed in collaboration with industry and research partners and based on real-life data sets (see Appendix A.2 for a list of the projects used). Project outcomes are then presented during a talk by all team members. The presentation as well as the Jupyter Notebooks created by the students during the work on their projects are used to assess student's success at the end of the lecture.

In the following, we will describe the content of both the lecture and the tutorial as well as the criteria for the final assessment in more detail. To this end, we follow an exemplary case study in which a large corpus of tweets is explored.

3 | PHASE I: LEARNING HOW TO PROGRAM

Learning how to code fulfills two goals in the context of this course: (a) students learn how to use the tools a programming language provides to analyze data. (b) learning how to program introduces students to algorithmic thinking [8], a core skill students should acquire through participation in this course.

The first phase of the lecture takes six weeks. The goal is to enable students with no background in programming to use Python for data analysis applications. Therefore we limit the content of the lecture to core features of a programming language that are useful for data analysis. More advanced programming paradigms such as object oriented programming or algorithm complexity are consciously left out. Phase (I) is accompanied by a series of learning videos for every core programming skill.

3.1 | Week 0

At the outset, we offer an obligatory lecture prior to the official start of the course that informs the students about the requirements and organizational aspects of the course as well as aiming to draw them into course by highlighting the importance of data analysis.

Starting out with the latter, we initiate the lecture by using the electronic voting system mVote [29] to survey the opinion of students regarding the level of importance of data analysis in our society. Subsequently, we ask one to three students of those indicating the highest level as well as those indicating the lowest level (if existent), why they thought that it was important. On the one hand this usually forebodes our arguments about the importance of data analysis while on the other hand it aims to instill an interactive atmosphere in the lecture as well as later in the tutorials. Subsequently we use three illustrative examples, relating data analysis with money making (via predictive advertising), health services (via case number analysis of cardiac arrests during the football World Cup in Germany) and love/sexuality (via illustrating the underlying logic of the Tinder match-algorithm). Last but not least, we turn towards organizational issues of the course (schedule, examination, etc.).

3.2 | Week 1

In the first week, students receive an introduction to Jupyter Notebooks [18], the programming environment they are going to use throughout the course. Access to Jupyter Notebooks is supplied centrally via a Jupyter Hub (https://jupyterhub.readthedocs.io/en/stable/), therefore students do not have to undergo a lengthy software installation process and can learn hands-on right from the start (see also supplement S1 for a more detailed description of the programming environment and technical implementation).

3.3 | Week 2

In the second week, students learn to use Python as a calculator. Different data types such as integers, floats and strings are introduced implicitly, along with the concept of variables. Students also have first contact with a function (`print()`), that allows them to inspect variables.

3.4 | Week 3

Subsequently, students are introduced to lists and loops. Loops are directly applied to inspect the content of a list of elements. The `pandas` [25] DataFrame is introduced as an extension of a list and basic container to store numerical data. This way, students are already introduced to the concept of programming libraries (such as pandas), that provide additional functionality. Data access via index and column name is practiced and basic statistics such as the sum and mean of elements are calculated.

3.5 | Week 4

In the fourth week, students learn simple logic operations such as testing whether a specific element is contained in a list. They are then introduced to conditionals (`if`, `else` and `elif`). Using conditionals they learn to create filters to access only a selected portion of data at a time. Students also create their first data visualization using `matplotlib` [15].

3.6 | Week 5

In the last week of phase one, students are formally introduced to functions in Python. Additionally, they learn how to build their own simple functions to automate repetitive steps and how different types of function arguments work in the context of functions from programming libraries. Additionally, students are introduced to non-numeric data types, namely images and text: they learn how to load, display and manipulate images and how to load and clean text.

Exercises for the tutorials re-iterate the concepts taught in the lecture and ask students to put them into practice. All exercises are provided as Jupyter Notebooks [23] in English and German language.

4 | PHASE II: CORE METHODS FOR DATA ANALYSIS

The second phase of the lecture takes four weeks. The goal is to teach the students core data science skills such as data acquisition, data cleaning, data exploration, data visualization and data analysis as well as introducing core concepts like probabilistic thinking by means of a data-driven learning process [11]. In the tutorials, students apply the skills conveyed in the lecture by working on a discipline-specific case study built around a contemporary data set. To this end, students are subdivided into smaller groups with similar domain backgrounds and work on different case studies - ideally touching upon their field of study. During the first run of our course we supplied three different case studies: an economics-related case study based on GDP and strike data [21], an archaeology-based case study based on data of pottery fragments found in the Mediterranean [20], and a general case study based on Twitter data [22]. All case studies follow a similar design and teach the same skills, except for a few domain specific applications such as visualization of data on a map (in the archaeology-based case study). In the following, we will illustrate the design of the case studies using the example of the Twitter case study which was used as a general-purpose case study.

4.1 | Week 6

In the lecture, the students are taught the underlying rationale and theoretical aspects of descriptive statistics - focusing in particular on measures of location, like the arithmetic mean and frequency tables. Using different historical examples the lecture also highlights the scope of skewing statistics one way or another by using different subsets or different presentation of results, ultimately pointing to the relevance of reflecting the origins of the data that is to be analyzed.

In the general purpose tutorial, the sources of the data are discussed. The Twitter data is compiled from three sources, featuring tweets of Russian trolls [7], Donald J. Trump [17] and regular Twitter users [12]. The data sets contain the tweet text, user account, tweet

language and a timestamp. The final goal of the case study is to find out if and how tweets from these three users or user groups are quantitatively different. Students are encouraged to inform themselves about the data sources, the authors and the context of the data collection. After the data is acquired, students explore the data sets by looking at the column names and data types and exemplary tweets. They then proceed to clean the data by identifying broken or unwanted entries (eg, tweets in the Trump data set that were not tweeted by the account of Trump) in the data set and removing these entries using filters. They also learn how to save a data set to disk, after they have finished their work on it.

4.2 | Week 7

During the second case study tutorial and the lecture, students learn how to visualize data. They are introduced to histograms to visualize statistics such as the tweet length and the number of words. To calculate the tweet length or number of words, they have to apply text processing techniques that were introduced in the lecture before. Students arrive at their first interesting finding, showing that the distribution of tweet lengths is very different between regular users and Russian trolls, and very optimized to a maximum number of 240 characters / tweet for Donald Trump. Students are also introduced to line plots which are useful to visualize time series, using the time stamp information contained in the data sets. While students try out different visualizations, they are introduced to styling and annotating plots using axis labels, titles and colors. Additionally students are asked to explore how information can be mis-represented by visualizations, for example by curtailing the axis ranges or choosing different bin-sizes for histograms. They are encouraged to look out for these kinds of mis-representations in public displays of quantitative information, such as newspapers.

4.3 | Week 8

In week 8, students are given deeper insights into descriptive statistics. They learn how to calculate the mean, median and SD of numerical data and what these measures mean. They combine this knowledge with a more domain specific exploration of the data set, such as a search for the number of hashtags or links used in every tweet, or the differences between tweet languages. They are also introduced to the idea of using these descriptive statistics to detect possible inconsistencies or outliers, such as a tweet with a very large number of characters, in the data set.

4.4 | Week 9

During the last week of the case study, students look closer into domain specific analysis of the data. Following an infamous YouTube Video [26] in which Donald Trump claims he “has the best words”, they try to find out how the words used by him are different from the words used by Russian trolls and regular Twitter users. To this end, students install a third-party library locally that allows them to check whether a word is a proper English word. They then proceed to filter the words and count the number of unique words used in the different data sets. Additionally, students are introduced to scatter plots to visualize the relation between tweet length and average word length. Finally, students learn how to perform a linear regression and quantify the relation between two variables (in this example tweet length and number of words used). Optionally, a short introduction to a Twitter data scraping tool is given to allow students to compile their own Twitter data sets and analyze them in the future.

5 | PHASE III: OUTLOOK AND PROJECT PHASE

Given the importance of data analysis in today's work inside and outside academia, phase III moves towards the practical application of those fundamentals by the students in form of a project. Work on the project is intended to yield an assessment for the participating students. Additionally, following [10], project work is intended to aid the students in their own contextualization of the scope, limits and meaning of their grasp of the course's content as well as giving a last, extensive example for application of the presented ideas that can help students to grasp them despite their often abstract nature [33].

Although the projects are the centerpiece of the third phase we additionally provide two optional lectures on regulatory aspects and further advanced methodologies regarding data analysis.

Regarding the projects, the students are aided by the tutors at the outset of the project to ensure that they initiate the work in time and are given the necessary aid to clear the first decisive obstacles before they are asked to complete their project during the semester break.

Ideas and data sets for the examination projects are compiled in collaboration with local companies and research groups. The main aim is to provide students with a realistic insight into data driven applications in industry and research that is based on real-life data sets.

All projects require the students to apply skills from four main areas to answer the project research question: (a) data cleaning and management, (b) visualization of information contained in the data and (c) calculation of descriptive statistics (d) inference using linear regression and correlation coefficients. These areas correspond to the previously given definition of "data acumen" [27]: ingesting data and thinking about how data processing might be affected by data issues (skills (b) and (d)) correspond to the practical task of data cleaning and management (1). During the calculation of descriptive statistics (3) and the answering of the research question using inference (4), students have to reflect on the analytical methods they employ (skill (e)). To complete the project, students have to combine code snippets and different programming libraries (skill (a)). For the presentation of the results, students will have to communicate about their workflow, data properties and issues and the approaches they employed (skill (f)). Choosing appropriate visualizations of data and results (area (2) and skill (c)) is also a dedicated part of the work of the project.

Some projects also include a data acquisition as subtask of (1). Many projects include additional optional questions which do not enter the final project assessment but guide students if they are willing to put in more work to follow their interests. The number of tasks involved in each of the four components differs between projects but is distributed in a way such that students can pass the class if they successfully complete all tasks from (1) and (2) and at least one task from (3). Additional completion of tasks from (3) and (4) awards better grades in the project assessment and project assessments uniquely determine the final grade students receive for the lecture. Work on the projects is done in groups of two to three students, aligning with the recommendations from [16]. The work is supposed to take approximately 80 hours per student and is partly completed during the semester break. The results of the project are presented to the lecturers and collaboration partners in the form of a short talk by all project group members and all group members receive the same mark. Materials needed to complete the projects, including data sets and code in the form of Jupyter Notebooks are also handed in at the end of the project. Projects undergo summative assessment based on the number of tasks students were able to complete and the quality of their presentation. In case of doubt of the originality of the student's work, the student's Jupyter Notebooks are consulted.

An exemplary project provided to us by a local organic farm was the analysis of the farm's energy consumption vs the energy produced by the farm's solar plant over the course of a year. The description of the project and task list as given to the students is included in supplement **(S2)**. The data was provided by the farmer in the form of several .csv files for the energy production of the plant and the energy consumption of the farm. The files were found to be disorganized, featuring corrupted and missing entries, as well as different time resolutions for different parameters. Therefore data management and cleaning was paramount. Regarding the analysis, the farmer was particularly interested in the economic viability of upgrading the solar plant by adding a storage facility. Accordingly, students were asked to characterize the solar plant's energy surplus during different seasons of the year and compare it to the farm's consumption and to evaluate the need for a storage facility. Students first cleaned and aggregated the provided data such that they had a single data table containing the timeline of all observables with the same time resolution. They then visualized timelines, calculated the energy surplus and interpreted the results (positive energy balance in summer and during days vs negative energy balance in winter and during nights). Using information on the cost of additional storage capacity in the form of batteries, they calculated the cost of a sufficiently large energy storage module to store enough energy during days/summers to supply energy during nights/winter. The students were able to show that the acquisition of additional storage capacity is not economically viable and therefore answer the main research question of the farmer. As an optional task the students were asked to prepare their analysis scripts in a user-friendly way for the farmer. Students were very interested in this question and were able to prototype a dashboard solution. Building onto this case study by the students a new research project has been initiated which explores the extension of the dash board by integrating an analysis of current weather data by means of deep learning algorithms to instruct the farmer regarding the use of energy intensive equipment in the upcoming 24 hours.

A second exemplary project description is outlined in supplement **S3**. This project required students to assess the weather dependent energy consumption of a seed drying facility. Unfortunately, because of data protection reasons, we cannot supply the original data for these projects.

An example for a project from the research field is about analyzing childhood malnutrition in Zambia. In this project, the students for instance used maps for visualizations as in Figure 1.

Regarding the lecture during Phase III, the format is more open with regards to topics, and contributions from other lecturers in the form of guest talks are welcome. Possible topic extensions are data ethics and algorithmic

bias, data protection, machine learning application and big data.

Both the projects and the additional lectures are thus geared towards using data analysis in practice with a particular focus on data cleaning, data visualization and description by metrics as well as the communication with and the presentation of the results to the collaboration partner. On the basis of Phase III, the students are guided into an active problem-tackling and problem-solving environment as described by [9]. Thereby, they are not only instructed with theoretical knowledge from lectures and custom made exercises but have experienced the difficulty of dealing with real-life data first hand and the challenges of molding the available data into the right format to address the commercial or research endeavors put before them as well as finally conveying the results of their work.

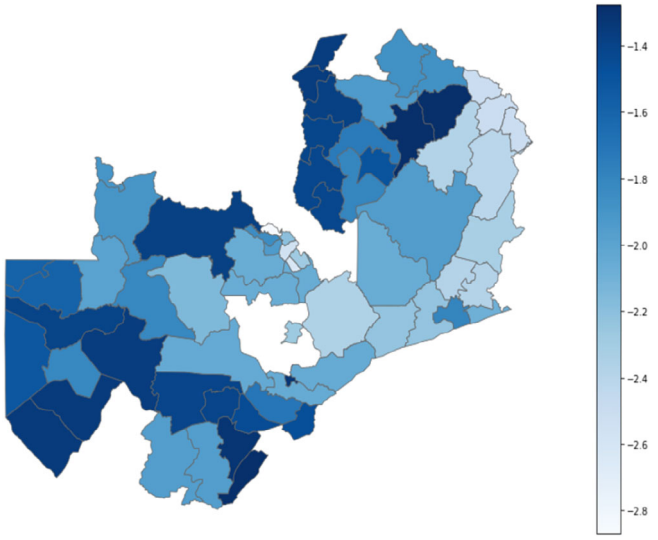

FIGURE 1 A map as a possible outcome of the student project on malnutrition in Zambia. The map shows the z-scores of the height of children up to the age of six for the different districts in Zambia [Colour figure can be viewed at wileyonlinelibrary.com]

6 | DISCUSSION AND LIMITATIONS

With the first implementation of our course we attracted 30 students, which is due to the fact that the course was offered as a purely elective module for the first year and was not fully integrated in the curricula of most faculty programmes yet. The majority (ie, 80%) of the participants where undergraduates in their bachelor's of which two thirds were in their fourth semester or below. Male and female participants were nearly balanced with 40% females and 60% males. Participants had very different backgrounds, ranging from Scandinavian studies over linguistics to history of economics. As illustrated in Figure 2, most participants had very little previous programming knowledge. Math and problem structuring skills were described as medium to high in the self-assessment administered before the first lecture.

Using a course design that teaches common skills to the plenum while splitting the participants into smaller, domain specific groups worked out very well and allowed students to use their domain specific intuition while applying newly learned skills in relation to data analysis. Students reported very high levels of interest (6.2 out of 7) and said they learned a lot in the course evaluation (5.8 out of 7). They reported that the workload they required was neither too low nor too high (4.5 out of 7) and gave a very good overall evaluation of the course (6.2 out of 7). In personal conversations students especially were particularly positive regarding the possibility to get in contact with companies and research groups during their exam projects. Equally, we got very positive feedback from the participating companies who pointed to their need of both practically minded and data-apt students, mirroring accounts on practically minded statistics courses elsewhere [4]. The digital teaching toolset and platform we chose, namely Jupyter Notebooks and a centrally administered Jupyter Hub instance, worked very well (see supplement S1 for a detailed description). This lead to all our students and

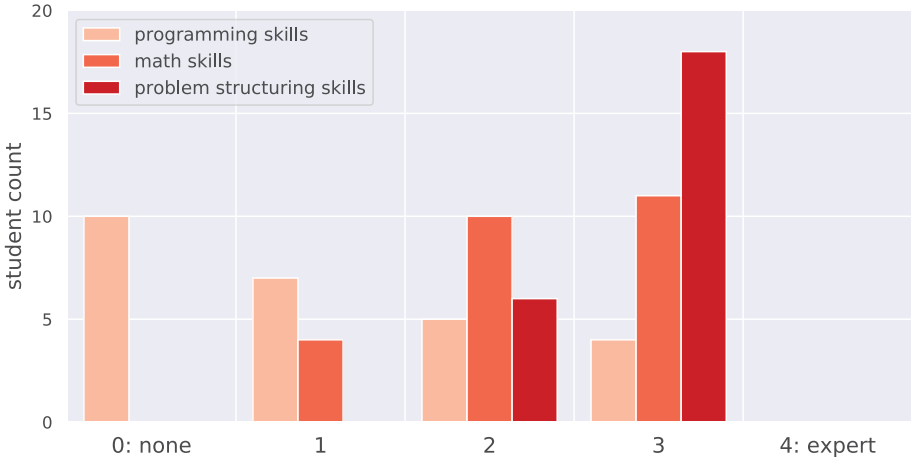

FIGURE 2 Self-assessment of skills by course participants before the start of the lecture. Skill level was assessed on a five point Likert scale, ranging from zero (no skills) to 4 (expert) for programming skills, maths skills, and problem structuring skills. Note: not all students participating in the lecture also participated in the self-assessment [Colour figure can be viewed at wileyonlinelibrary.com]

tutors having access to an uniform and well-functioning computing environment without spending much personnel resources. In addition, these tools are capable of scaling up to 100's of students with minimal effort.

A limitation of our approach is certainly the rather high number of tutors needed to supervise the students in the tutorials. Every tutorial had two tutors, where at least one had advanced programming and statistics knowledge. For the second tutor position we preferentially selected tutors with domain knowledge for the specific tutorial. Initially we planned for an average of 15 students per tutorial. Given the initial turnout of 30 students, the number of students per tutorial ranged between 4 and 10, yielding a somewhat problematically high tutor-student ratio.

Another limitation of the course design is the high amount of work required to design several parallel and domain specific case studies. This was possible because we received a grant to develop teaching materials. We hope that by making our teaching materials openly accessible, we can lower the barrier to teach a similarly styled course for other colleges and universities.

There is a broad consensus ranging from human resources departments in Germany to the international literature on teaching statistics regarding not only the eminent importance of data competencies but also the need to provide students with practical hands-on experience of applying basic statistical tool sets in practice [4,11,16].

Given the growing need for such competencies among the full breadth of students graduating German universities, new teaching formats such as the one documented here are in need - as is an intensified discussion on the conveyed statistical content, the datasets provided and the didactic methods employed to improve these much needed courses further.

ACKNOWLEDGEMENTS

The authors thank the Stifterverband and the Heinz-Nixdorf foundation for providing the funding for this work. Open access funding enabled and organized by Projekt DEAL. WOA Institution: GEORG-AUGUST-UNIVERSITAET GOTTINGEN Blended DEAL: ProjektDEAL.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

ORCID

Jana Lasser https://orcid.org/0000-0002-4274-4580
Alexander Silbersdorff https://orcid.org/0000-0002-3453-9536
Benjamin Säfken https://orcid.org/0000-0003-4702-3333
Thomas Kneib https://orcid.org/0000-0003-3390-0972

REFERENCES

1. Muhammad Ateeq, Hina Habib, Adnan Umer, and Muzammil Ul Rehman, *C++ or python? which one to begin with: A learner's perspective*, 2014 International Conference on Teaching and Learning in Computing and Engineering, IEEE, New York City, 2014, pp. 64–69.
2. Dilhari Attygalle and Asoka Ramanayake, *Statistics in practice: making of professional statisticians in a classroom*, Proceedings of the 10th International Conference on Teaching Statistics, International Statistical Institute, Voorburg, The Netherlands, 2018.
3. Ben Baumer, *A data science course for undergraduates: Thinking with data*, Am. Statist. 69(4) (2015), 334–342.
4. Brian Jersky. *Statistical consulting with undergraduates–a community outreach approach*, Proceedings of the 6th International Conference on Teaching Statistics, International Association for Statistical Education, International Statistical Institute. Voorburg, 2002.
5. Theodore Chadjipadelis and Ioannis Andreadis, *Use of projects for teaching social statistics: case study*, Proceedings of the 8th International Conference on Teaching Statistics, International Statistical Institute, Voorburg, The Netherlands, 2006.
6. Bruno de Sousa and Dulce Gomes, *Teaching statistics using R at a college or a university level: it can be possible?* Proceedings of the 10th International Conference on Teaching Statistics, International Statistical Institute, Voorburg, The Netherlands, 2018.
7. FiveThirtyEight. *Tweets from russian trolls*, available at https://github.com/fivethirtyeight/russian-troll-tweets/.
8. Gerald Futschek, *Algorithmic thinking: the key for understanding computer science*, International conference on informatics in secondary schools-evolution and perspectives, Springer, Berlin Heidelberg New York, 2006, pp. 159–168.
9. Iddo Gal, Lynda Ginsburg, and Candace Schau, *Monitoring attitudes and beliefs in statistics education*, Assessm Challenge Statist Educat 12 (1997), 37–51.
10. Joan Garfield, *How students learn statistics*, Int. Statist. Rev. 63 (1) (1995), 25–34.
11. KS Gibbons and Helen MacGillivray, Education for a workplace statistician, Topics from Australian Conferences on Teaching Statistics, Springer, Berlin Heidelberg New York, 2014, pp. 267–293.
12. Alec Go, Richa Bhayani, and Lei Huang, *Tweets from regular twitter users*, available at http://help.sentiment140.com/for-students/.
13. J. Hardin et al., *Data science in statistics curricula: Preparing students to "think with data"*, Am Statist 69(4) (2015), 343–353.
14. Ambikesh Jayal et al., *Python for teaching introductory programming: A quantitative evaluation*, Innovat. Teach. Learn. Informat. Comput. Sci. 10(1) (2011), 86–90.
15. D. Hunter John, *Matplotlib: A 2d graphics environment*, Comput. Sci. Eng. 9(3) (2007), 90–95.
16. Katherine Taylor Halvorsen, *Formulating statistical quiestions and implementing statistics projects in an introductory applied statistics course*, Proceedings of the 8th International Conference on Teaching Statistics, International Statistical Institute, Voorburg, The Netherlands, 2010.

17. Michael W. Kearny, Tweets from Donald J. Trump, avaiable at https://github.com/mkearney/trumptweets.
18. Thomas Kluyver et al., in Jupyter notebooks – a publishing format for reproducible computational workflows, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, F. Loizides and B. Schmidt, Eds., IOS Press, Amsterdam, 2016, 87–90.
19. Theodora Koulouri, Stanislao Lauria, and Robert D. Macredie, *Teaching introductory programming: A quantitative evaluation of different approaches*, ACM Trans. Comput. Educat. (TOCE) 14(4) (2014), 1–28.
20. Jana Lasser and Debsankha Manik, *Archaeology case study*, available at https://github.com/Daten-Lesen-Lernen/daten-lesen-lernen-lecture/tree/master/case-study-archaeologie.
21. Jana Lasser and Debsankha Manik, *Economy case study*, available at https://github.com/Daten-Lesen-Lernen/daten-lesen-lernen-lecture/tree/master/case-study-wirtschaftswissenschaften.
22. Jana Lasser and Debsankha Manik, *General purpose case study*, available at https://github.com/Daten-Lesen-Lernen/daten-lesen-lernen-lecture/tree/master/case-study-allgemein.
23. Jana Lasser and Debsankha Manik, *Lecture "Daten Lesen Lernen"*, avaiable at https://github.com/Daten-Lesen-Lernen/daten-lesen-lernen-lecture.
24. Shanhong Liu, *Most used languages among software developers globally*, Statistia, London, Vol 2020, 2020, 06.
25. Wes McKinney. *Data structures for statistical computing in python*, Proceedings of the 9th Python in Science Conference, SciPy Organizers, Austin, Texas, 2010, pp. 51–56.
26. MSNBC (2018). President Donald Trump: 'I Have The Best Words' All In MSNBC [video]. Retrieved from. https://www.youtube.com/watch?v=lM2GFtO5VP0.
27. National Academies of Sciences, Engineering, and Medicine and others, *Data science for undergraduates: Opportunities and options*, National Academies Press, Washington D.C., 2018.
28. Fernando Perez, Brian E. Granger, and John D. Hunter, *Python: an ecosystem for scientific computing*, Comput. Sci. Eng. 13(2) (2010), 13–21.
29. Almut Reiners, Sebastian Hobert, and Matthias Schumann, *Lernen mit Smartphones an der Georgia-Augusta-eine Zwischenbilanz*, DeLFI Workshops, Bonn, 2014, 180–188.
30. Chantel Ridsdale, James Rothwell, Michael Smit, Hossam Ali-Hassan, Michael Bliemel, Dean Irvine, Daniel Kelley, Stan Matwin, and Bradley Wuetherick, Strategies and best practices for data literacy education: Knowledge synthesis report, Tech. report, 2015.
31. David Robinson, *The Incredible Growth of Python | Stack Overflow*, Stack Overflow, New York, 2017.
32. Rob Root and Trisha Thorme, *Community-based projects in applied statistics*, Am. Statist. 55(4) (2001), 326–331.
33. Jon Singer et al., *Constructing extended inquiry projects: Curriculum materials for science education reform*, Educat. Psycholog. 35(3) (2000), 165–178.
34. Charles C Taylor, *Using R to Teach Statistics*, Proceedings of the 10th International Conference on Teaching Statistics, International Statistical Institute, Voorburg, The Netherlands, 2018.
35. Ian Westbrooke and Maheswaran Rohan, Statistical training in the workplace, *Topics from Australian Conferences on Teaching Statistics*, Springer, New York, 2014, 311–327.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of this article.

How to cite this article: Lasser J, Manik D, Silbersdorff A, Säfken B, Kneib T. Introductory data science across disciplines, using Python, case studies, and industry consulting projects. *Teaching Statistics*. 2021;43:S190–S200. https://doi.org/10.1111/test.12243

APPENDIX: A.LIST OF SUBJECT-SPECIFIC CASE STUDIES AND PROJECTS

List of subject-specific case studies

The three subject-specific tutorial threads offered in the summer semester 2019 were:

A thread aimed at students from linguistic fields using data sets from Twitter. This thread was also used as a the default thread for students from fields not addressed by the other three subject-specific threads.

A thread aimed at economics students and students from (modern) history-related fields considering data regarding domestic production and industrial action in several countries in the past.

A thread considering pottery-origins aimed at students from archaeology and (ancient) history-related fields.

The students were free to choose, which thread to attend, but generally speaking the provisioned topic-subject allocation was as expected.

List of subject-specific projects

The twelve subject-specific projects offered in the summer semester 2019 were:

A project from an IT-company enquiring about insolvency rates in different industries to evaluate credit risk.

A second project from the same company requiring a match of customer files with public offense registers.

A project from a retail company regarding customised advertising.

A project from an ecological farm regarding a solar power plant.

A project from a company focused on plant-breeding regarding agricultural production in different countries.

A second project from the same company considering the drying in a production process.

A third project from the same company considering measurements regarding the breeding of plant-hybrids.

A project from an education start-up to evaluate cooperation potential between the local universities and regional companies.

A project from an engineering company regarding predictive maintenance.

A project from a regional infrastructure initiative regarding the organisation of bus routes.

A project from a research-group regarding ancient pottery.

A project from another research-group regarding malnutrition among children.