BigSurv

J. Kasmire

September 2020

Abstract

Computational social science (CSS) requires a unique blend of knowing how to do social science research and knowing how to do computationally intensive research. These are, in some interesting ways, very difficult skills to blend. Social science research demands an ability to think like a human, with abstract concepts, context-dependence, inference and awareness of the probabilities of shared background knowledge. In contrast, computationally intensive research demands an ability to think like a computer, with concrete terms, absolutes, hierarchies, and potentially unique frames of reference. Thus, CSS requires a capacity to think in multiple ways that are at best occasionally difficult to combine and at worst fundamentally incompatible. For example, social scientists might a variety of traditional social science methods to research fuzzy and very human concepts like "trust". To do so, they could use natural languages, semi-structured processes and multiple theoretical frameworks, all without needing a singular or universal definition of "trust". However, were they to use computational methods, they would have to explicitly define how trust would be represented in a computer-readable format, perhaps as a number between 0 and 100, between –1 and 1, or as a statistical model with multiple factors, each measured numerically. Further, they might need to explicitly define what is understood as a "default level" of trust alongside clear rules by which that default value changes over time as a consequence of behaviour or interaction.

The disparate CSS skills present a challenge; those drawn to social science by an affinity for abstract concepts, human communication, context-dependence or applying generalised societal knowledge might struggle to gain or use computational research skills. At the same time, those drawn to computational research through an intuitive grasp of how information can be structured, organised, or manipulated might become very frustrated by research topics that have ill-defined boundaries, that change meaning in different contexts, or that rely on assumptions or shared experience for interpretation.

Although challenging, this dichotomy can be at least partially resolved through careful training that fully acknowledges how some trainees may feel some concepts to be blindingly obvious while others are totally impenetrable. This paper compares and contrasts the two independent skill sets needed for CSS and highlights why researchers may be unevenly adept at each, especially if they have already established a history of work in one or the other. It reviews the evidence underpinning pedagogical approaches for developing computational skills, presents a framework for training future computational social scientists in the basics of both skill sets, and lays out some considerations for further training and skill development.

1 Introduction

In the modern world, people are spending more time communicating, learning, working, managing daily tasks and generally living their lives through the medium of computers. This not only shapes those interactions and behaviours but also leaves records that are unprecedented in both detail and volume. Studying digitally mediated peoples' behaviour through the data generated by those interactions or through other computational means requires researchers to have a good grasp of both how humans and how computers work and communicate. Traditionally, social scientists are trained to understand humans but not computers and computer scientists are trained to understand computers but not humans. Thus, the very different skills and methods taught to each means that neither traditional social science nor computational science is entirely well placed to study the digitally mediated interactions and behaviours of people.

Consequently, a new approach is needed to take advantage of the increasing digitalisation of modern life [1]. Computationally skilled social scientists, or computational social scientists, would be able to identify important questions of how people are changing in response to the growing reliance on and access to computers. At the same time, they would also be able to identify new digital resources or computational methods that can shed light on more traditional social science questions about how people interact and behave. Thus, CSS presents both an opportunity and a challenge.

In the short term, the challenge will be to train social scientists and computer scientists in the skill sets and perspectives to which they have not already been exposed. As each is likely to have been drawn to their respective fields by natural inclination, learning to think differently and use new methods may be potentially uncomfortable. However, the challenge over a longer term is to develop a new field of CSS that trains researchers in all of the skills they will need, which embeds the ability to think about both humans and computers, and which encourages good working practices so that the results of their research can be communicated effectively with multiple audiences.

2 Necessary skills

CSS requires all the skills needed for social science as well as additional skills from computer or data sciences. These are not particularly easy skills to blend as one requires thinking like people and the other requires thinking like computers. This section compares and contrasts the two skills sets and comments on how they can be used together to do CSS effectively.

2.1 Social Science skills

Social science research studies people, societies and the relationships and behaviours of people within societies and can include economics, linguistics, archaeology, business studies, psychology, geography and more. Most of these disciplines require that researchers be able to understand and think like people, which means using (or at least being tolerant of) abstract concepts, fuzzy or overlapping categories, inference and awareness of the importance of shared knowledge and context-dependence. This allows researchers to identify important research questions or problems to address and to justify why addressing them would have a significant social or economic impact.

A series of bench-marking reviews conducted by the Economic and Social Research Council in the UK found that there were generally good levels of skills among most social scientists [2], including information literacy or library skills [3], qualitative research skills and quantitative research skills. Thus, social scientists are good at collecting, categorising, synthesising and analysing complex and disparate information, detecting patterns, drawing inferences, supporting arguments, and communicating results. However, that ESRC review found that some social science sub-disciplines were comparatively weak in quantitative analysis skills [2], more widely known as the 'quantitative deficit' [4, 5]. Importantly, social science students entering doctoral study had low quantitative skills which not only required time and effort to remedy but which tended to steer their research toward qualitative approaches [6]

It seems that social sciences in the UK have emphasised a 'critical', rather than an 'analytical' approach and so have not traditionally been strong on quantitative and empirical methods including surveys, experimental methods, and collection and analysis of primary or secondary data [7]. The quantitative methods taught to social science students are often limited to statistics and statistically modelling, many of include simplifications or linear assumptions that may not be a good fit for the complex, dynamic and potentially chaotic nature of human societies [4]. Moreover, the limited quantitative skills typically taught to social science students are almost always presented in 'student friendly' function-specific software tools rather than more broadly applicable coding languages. This can allow learning the specific software to takes precedence over learning methodological logic [8, 9] unless learners are instructed in how to translate research questions into quantitative and empirical research methods [10].

Thus, CSS is not simply social science plus numeracy. Instead, CSS must create a new philosophy of what social sciences are and how they can be done [11, 4, 5, 7]. This will mean asking those who can think like people and ask important questions about people to learn how to think like computers so they can answer those questions about people in new ways using computational methods and analyses.

2.2 Computer Science Skills

Computer science research studies the theoretical and practical application of algorithms, computing and information processes such as control, perception, intelligence and learning [12, 13]. Computer science requires that researchers be able to understand and think like computers, meaning that they must be comfortable with how computers use concrete definitions, absolute categories, strict hierarchies, formal structures and they have no access to information or processes that are not clearly defined within the appropriate scope. This allows researchers to employ programming scripts, computational methods and technological tools to collect, manipulate, and analyse data to address informational problems and to improve computational processes.

In a world that is increasingly digital, computer scientists are in high demand. Success in computer science requires obvious skills like technical knowledge, reasoning, problem-solving and project management but also less obvious skills like creative thinking, personal resilience, persuasion, interpersonal communication and integrity [14]. A review of the state of UK computer science finds that employers are diverse, although more likely to be in business or industry than in research, and that they broadly support how computer science graduates are trained in fundamental principles [15]. At the same time, the review highlights that many employers are not content with the 'soft skills' and practical experience gained by students during a computer science degree [15], suggesting that courses may not be entirely balanced. Traditional computer science courses can be too focused on the passive acquisition of technical skills with relatively little attention given to teaching, or even encouraging the use of, critical reflection, creativity, interaction, or communication [16]. In effect, graduates are not always able to identify valuable research concepts or themes, justify research decisions or interpret and discuss research conclusions effectively, even though these are all valuable 'soft skills' that computer scientists need [14]. Computer science courses can be restructured to focus on the principles of well motivated research, critical thematic analysis, active engagement and presentation skills led to more balanced skill acquisition without sacrificing the technical content of courses [16], but this does not appear to be commonplace. Anecdotal experience suggests that even educators with good reputations in computer science do not provide a good range of practical experience and do not show graduates how they can apply the knowledge they have gained in the course.

2.3 Example - Trust

An admittedly extreme example shows how differently social scientists and computer scientists might approach the same topic of how to measure and explore 'trust'. A hypothetical social scientists might want to explore whether there are important patterns that define how citizens come to trust public figures. To investigate, they could first read up on definitions of and frameworks for interpreting trust in order to define trust as it applies to their research question. They might then use a series of semi-structured interviews in which people are asked how trust is developed or lost, whether they find some people easier to trust than others, and other similar qualitative questions. All of this might then be used to develop a measure of trust, after which a the social scientists might conduct a second round of interviews, or use a survey or a questionnaire, in which certain famous people are scored as 'trusted' or not according to the newly derived measure of trust.

In contrast, a computer scientist might want to explore how consumers use digital marketplace seller ratings when choosing products in order to reduce expensive complaints or product returns. They are less likely to research theories or frameworks of trust, but might research algorithms behind how the ratings are created or updated, would probably define trust numerically (e.g. a number between 0 and 100 or between -1 and 1), and might develop alternative rating algorithms, user experience designs that emphasise the ratings, or ways to promote sellers with higher ratings. These alternatives algorithms, user experience designs or seller promotion methods could be tested in beta or in selected markets, with the impacts tracked by changes to complaints or product returns.

3 Computational social science

Social science and computer science are both important, not least of which because they seek to answer very different kinds of problems and use very different methods to do so. However, neither of them is entirely well placed to research modern interactions and behaviours that take place on or are tracked by digital media. Studying these requires social science skills and the ability to think like people in order to identify the important problems, consider possible solutions, and connect with relevant information. At the same time, researchers will need computer science skills and the ability to think like computers in order to access and work with large volumes of (complex) data in efficient, accurate and reproducible ways. These mixed problems will only grow more common as resources and records are increasingly digitised, interactions, objects and processes become 'smart' or network connected, large volumes of data become available or are updated at ever faster rates, and many more ways that life may become more reliant on computation.

Combining the skills of social scientists and computer scientists allows computational social scientists to:

- process and analyse large volumes of unstructured data in ways that were previously prohibitively time-consuming,
- capture data generated or published in real-time to avoid significant delays on problems that deserve rapid action [17],
- access information on new activities such as social media use or on previously unmeasurable activities such as minute-by-minute energy use,
- combine and work with existing data on unprecedented scales or with greater dynamism and complexity in order to gather new insights or create

new visualisations [18], and

- potentially much more.

Obviously, a computational social scientists would want both skill sets, but this is not trivial as individuals are likely to be unevenly adept at each. Their level in each of the disparate skills sets would depend on their training background, personal inclinations, the dominant methods used in their research environment, and other factors. Those individuals who were originally drawn to social science by an affinity for abstract concepts, human communication, context-dependence or applying generalised societal knowledge might struggle to gain or use computational research skills that depend on strict rules, exclusive definitions, or extremely formal and structured processes [19]. Similarly, those drawn to computer or data science through an intuitive grasp of how information can be logically organised, accessed and manipulated might become very frustrated by working with the relatively ill-defined, overlapping, context-dependent concepts common to social science as well as the reliance on assumption and background knowledge needed for interpretation.

But it is not enough to take someone from one skill set and simply add more skills from the other. To explain, consider the concept of programmes and programming. These are commonly, but incorrectly, understood to be strictly computational concepts. In fact, a very large number of things in very diverse media can be understood as programmes as long as they are ways of "describing processes of production" and "convey a particular aesthetic" [11]. Thus, knitting patterns, architectural blueprints, recipes, musical notation and much more can all be interpreted as programmes that describe how to do or make a particular thing in a particular way. Importantly, a programme is an attempt to effectively and accurately describe a process of production. A good programme must be both very understandable and capable of producing the desired thing with sufficient accuracy.

Using this logic, CSS should go beyond simply answering social science questions using computational, quantitative or empirical methods and instead aspire to recreate the philosophy underpinning social science [11, 4, 5, 7, 20]. This means that CSS should approach those vital but nebulous social science questions with a 'programming mindset' so that the projects, research questions, methods and concepts (both traditional and computational) are transparent, useful and reproducible [20, 4]. There is a clear need for and value to be gained from embedding a programming mindset into social science to create CSS. Society can only benefit from addressing its most important questions through critical, analytical, qualitative, quantitative and empirical research that is also well communicated, transparent and reproducible. Those working in CSS will need to think like people and like computers so that important questions can be answered in ways that are people can understand, trust, validate and use. This will not be easy.

4 Resolving dichotomy: a proposed framework

Acquiring all of the skills needed to be a CSS presents a challenge. While a significant challenge, the best approach seems to be through education and training. In the short term, this will require that social scientists try to acquire a programming mindset and computer scientists gain the 'soft skills' that social science researchers are trained in. Regardless of what discipline they started in, both will need to be able to switch between thinking like humans and thinking like computers to translate research questions into computational methods and then translate results into effective science communication.

But over the long term, and especially as CSS becomes recognised and valued, new researchers can be trained to use an innovative proposed framework. This framework can be understood as a scientific method for CSS computational social scientific method and consists of eight steps. Each of these steps is explained briefly below and is accompanied by a example of how the step might play out in a theoretical CSS research project.

1. **Problem identification** Researchers can start by either asking a question and looking for data to address it or by exploring a new data for some pattern that needs explanation. Regardless of how the research process starts, the problem must be clearly identified, which means being as specific as possible, about what question needs to be addressed, what pattern needs explaining, what insight is missing, or what response is interesting. This should include details about the time frame, scope and scale of the problem, as well as who the problem belongs to, who or what else is involved, and what role the researcher will play in addressing the problem.

   Example Upon noticing that '#pizzagate' had been spray painted on a wall in her area, a researcher decides to investigate how information transmitted between individuals is linked to subsequent criminal behaviour. To properly identify this problem, she needs to specify if she is interested narrowly in '#pizzagate' graffiti or more widely in criminal damage related to online conspiracies, such as damage to phone masts following speculation about 5G as a source of new viruses. She would need to define a time frame for study, what media sources to consider, locations for criminal damage, and more. She might see police forces as the problem owners with social media companies, news outlets, community groups and local governments as relevant stakeholders and would see her role in the research process as creating a tool to predict increases in criminal damage stemming from online activity.

2. **System identification and decomposition** After identifying the problem, the researchers must gather relevant information and perspectives on the system in which the problem is found and what that system is composed of. This could be through traditional social science research methods like surveys, observation, interviews or literature reviews, but it could also be through preliminary computational data collection via

web-scraping, API integration, corpora building, app creation, and more. During the information other appropriate sources of insight on the problem. Researchers must also carefully spell out any sub-problems, processes, relationships, simplifications or assumptions that are revealed in the information gathering process.

Example Continuing with the previous example, the researcher would probably need to speak to experts or read literature on conspiracy theories, information transmission, vandalism, criminal behaviour and group think, among other topics. She would need to identify one or more online conspiracies to study and how information about these can be identified through keywords, hashtags, or original posters as well as how vandalism and criminal damage is identified, reported and categorised and whether descriptions or photos of that damage are accessible. She would probably try to map networks of individuals who are exposed to or that have shared information on the chosen conspiracies and what other groups they belong to as well as identify the time scales covering the creation and distribution of information. She would also need to clarify any assumptions, such as what effect vandals expect their effects to have.

3. **Concept formalisation** Although the relevant concepts may already seem well defined, computers are ill-equipped to deal with any ambiguity, context dependency, or natural language terms. This means that all of the concepts and processes identified so far, as well as hypotheses derived, must be made explicit, formal, and both computer- and human-understandable. In many cases, this step will include pseudo-code to spell out how the concepts might look when formalised in computational methods.

   Example The researcher would need to define exactly what data she will collect and how it will be collected (e.g. the date, user name, rough geographic area and contents of tweets with keywords or hashtags via an API) as well as how it will be processed, stored and managed. She would also need to clarify how hypotheses will be tested through data analysis, for instance, by creating a predictive model that relates hashtag popularity, longevity or trending status to reports of vandalism and criminal damage.

4. **Data collection, software implementation and verification** Researchers must select and implement an appropriate computational method or combination of methods to address the problem, which will be highly dependent on the research specifics. Whatever methods are chosen or software is used, the implementation needs to be explicitly verified to ensure that conceptualisation was correctly translated into software, as computers do exactly what they are told to do, which might not be the same as what they were meant to do.

   Example The researcher would need to write code to acquire and process the data and build the predictive model. Importantly, the code would need to be thoroughly tested and verified to ensure that it is capturing as much of the correct data as possible without too much unrelated data (e.g. catching #pizzagate, #PizzaGate and #PiZaGaTe but not #pizza, #gate or #pizzagrate) and that it is processing as intended (e.g. ensuring that dates from tweet and police reports are in the same).

5. **Experimentation and/or data analysis** Obviously, researchers will need to run their experiments, create their models, build their simulations, analyse their data or otherwise implement the computational methods that they have chosen. They will also need to identify and visualise patterns in the results, and interpret these to explain outcomes in relation to the original problem and conceptualisation.

   Example Continuing the given example, the researcher would need to acquire and process the data, build the predictive model, analyse the model results and interpret the outcomes. This might include creating visualisations, defining the model accuracy and precision, identifying under which conditions the model works best and worst, and many more.

6. **Discussion, policy recommendations, etc.** Beyond identifying and explaining the results, the researchers must also draw conclusions about how the work relates to the original problem, how they reveal further problems, or what future research might need to be done. Again, this will be highly specific to the problem under investigation, but also to the results of the experimentation and data analysis stage.

   Example Assuming that the predictive model was sufficiently accurate and precise under at least some conditions, the researcher would need to explain the results in non-specialist terms and how they mightw shape policy, change police reporting processes or otherwise be turned into a real world impact. She would also need to comment on the limitations of the model as it stands, how it might be improved and whether the improvements might allow it to apply more generally to other conspiracies or even other online behaviour.

7. **Communication, publication, and presentation** All of the above must then be clearly communicated to scientific and lay audiences through multiple means, not only to draw attention and promote the research but to affect change in the real-world problem the research set out to address. This applies equally to the results related to the research question and to the research methods used to obtain those results because answers are very hard to believe if it is not clear how that answer was obtained. The communication changes over time as short term publication and long term engagement require different foci.

   Example The researcher would almost certainly want to publish the results in academic journals and conferences, but would probably also want to ensure that the work features in a press release. Beyond that, the researcher would want to consider how the research could be made widely accessible (e.g. an interactive website or app), how it could be brought to diverse audiences (through blogs, tweets, podcasts, science festivals, etc.) and how to ensure that any policy changes or impacts of the work and reported over the longer term.

8. **Data sharing, documentation and validation** While verification ensures the work was done right, validation checks the right work was done in a convincing and reproducible way. This is not unique to CSS, but a transparent and well documented workflow allows other researchers to understand and validate the research. Ideally, this would include providing access to code and data when possible.

   Example The researcher should strive to take good notes throughout all the entire research process, being especially clear about which decisions were made and why, what data was used, how it was processed, what methods were applied and how the results were analysed. She might keep all of the data, code and notes in a publicly accessible, version-controlled repository (e.g. github). If the data and/or code are too sensitive to release in full, then the researcher might make a synthetic version of the data set or an extremely detailed version of the code and research methods available instead.

Listing these steps in order, especially when enumerated, suggests that the steps move forward in a clear linear progression; in practice, the steps are less clearly distinct and each step can reveal the need for new passes through previous steps. If the verification step shows that the software implementation is not functioning properly, then researchers would need to go back and rework the code. They may even have to go back to the system decomposition or concept formalisation stages to check that deep-seated errors, assumptions or counterproductive ideas are not preventing the software from working properly. Even the first step, that of identifying the problem, could come after data collection if existing data is explored with an open mind.

As well as some steps being iterative, some are best seen as parallel. Specifically, data sharing, documentation and validation is listed as the final step but should actually apply throughout the entire research process. Everything from ideas and thoughts on during the initial problem identification phase, right through to communicating the results should be well documented so that other researchers can properly understand the work, can reproduce it, explore the same question through other methods, apply the methods to other questions or use and modify the method as needed. This is an essential part of how science works and how scientific knowledge accumulates and changes over time. Importantly for CSS, this is also a key feature of how new research methods come to be accepted and used more widely within research.

5 Future considerations

Clearly, this framework is only the beginning of developing a best practice for how to train computational social scientists. It does not include all of the specific skills that computational social scientists may need to know. Indeed, no framework could hope to include all the skills they might need as these will almost certainly increase more rapidly than educational programmes can keep up. However, there are a few obvious questions that computational social scientists might want to explore.

- What is a good age to learn the different CSS skills? Language, for example, has a window of opportunity for learning, after which it is very hard for learners to become fluent. The same may be true of learning to think like people or like computers, which would mean that people need to have some exposure to both kinds of thinking before that window closes if they are to become proficient in both.

- Does how CSS skills are learned matter? There may be quantifiable differences in the way that individuals are able to employ different skills depending on the order in which skills are learned, how the skills are taught, or similar.

- What is an acceptable minimum level of each skill set for effective collaboration on CSS research projects? Very few individuals are likely to be equally proficient in all of the skills needed for a CSS project, and collaboration seems to be increasingly important in research. Thus, CSS education and training should anticipate that individuals may want to specialise on some skills more than others but that effective communication with others, including non-computational social scientists, will be needed.

- What are good ways to balance generalist and specialist skills in collaborative projects? Although individuals are likely to specialise to some extent, care must be taken that computational social scientists balance their natural inclinations with the specific benefits of having a balanced skill set. Failure to do so would leave a collaborative project at risk of misunderstanding or even complete failure to start, as is currently the case with many projects in which highly technical and non-technical researchers try to collaborate.

- Are there other steps to include in a computational social scientific method or other skills that need to be taught? Over time, this framework is expected to develop, with some steps added, removed or split to account for new insights into best practice. Some of the most likely changes may already be known, or at least considered.

References

[1] Susan Halford and Mike Savage. Speaking sociologically with big data: Symphonic social science and the future for big data research. *Sociology*, 51(6):1132–1148, 2017.

[2] Paul Simmonds and Lark Parker-Rhodes. Evaluation of the ESRC International Benchmarking Review Series. Technical report, Economic and Social Research Counci, 03 2011.

[3] Schubert Foo, Xue Zhang, Yun-Ke Chang, Shaheen Majid, Intan Azura Mokhtar, Joanna Sin, and Yin-Leng Theng. Information literacy skills of humanities, arts, and social science tertiary students in singapore. *Reference & User Services Quarterly*, 53(1):40–51, 2013.

[4] David Byrne. Uk sociology and quantitative methods: Are we as weak as they think? or are they barking up the wrong tree? *Sociology*, 46(1):13–24, 2012.

[5] Malcolm Williams, Luke Sloan, and Charlotte Brookfield. A tale of two sociologies: Analyzing versus critique in uk sociology. *Sociological Research Online*, 22(4):132–151, 2017.

[6] John MacInnes. Quantitative Methods Strategic Advisor report. Technical report, Economic and Social Research Counci, 01 2015.

[7] Malcolm Williams, Luke Sloan, and Charlotte Brookfield. The quantitative crisis in uk sociology. *Data in Society: Challenging Statistics in an Age of Globalisation*, page 337, 2019.

[8] Jennifer Buckley, Mark Brown, Stephanie Thomson, Wendy Olsen, and Jackie Carter. Embedding quantitative skills into the social science curriculum: Case studies from manchester. *International Journal of Social Research Methodology*, 18(5):495–510, 2015.

[9] Julie Scott Jones and John E Goldring. I'm not a quants person'; key strategies in building competence and confidence in staff who teach quantitative research methods. *International Journal of Social Research Methodology*, 18(5):479–494, 2015.

[10] Christina Silver and Nicholas H Woolf. From guided-instruction to facilitation of learning: the development of five-level qda as a caqdas pedagogy that explicates the practices of expert users. *International Journal of Social Research Methodology*, 18(5):527–543, 2015.

[11] Simon Yuill. Programming as practice. In J. Gibbons and K. Winwood, editors, *Hothaus Papers: perspectives and paradigms in media arts*, pages 87–95. ARTicle Press, Birmingham, 2005.

[12] Peter J Denning. Computer science: The discipline. *Encyclopedia of computer science*, 32(1):9–23, 2000.

[13] Luciano Floridi. *Information: A very short introduction*. OUP Oxford, 2010.

[14] Rajnish Kumar Misra and Khushbu Khurana. Employability skills among information technology professionals: A literature review. *Procedia computer science*, 122:63–70, 2017.

[15] Nigel Shadbolt. Review of computer sciences degree accreditation and graduate employability, department for business, innovation & skills and higher education funding council for england, 2016.

[16] Lori Pollock. Integrating an intensive experience with communication skills development into a computer science course. In *Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education*, pages 287–291, 2001.

[17] Noortje Marres and Esther Weltevrede. Scraping the social? issues in live social research. *Journal of cultural economy*, 6(3):313–335, 2013.

[18] Mark Carrigan and Will Housley. Social science & social futures: Fast scholarship, emengying technologies & the future of inquiry. hrefhttps://discoversociety.org/2017/01/03/social-science-social-futures-fast-scholarship-emerging-technologies-the-future-of-inquiry/, 2017.

[19] Tom Jewett and Rob Kling. The dynamics of computerization in a social science research team: A case study of infrastructure, strategies, and skills. *Social Science Computer Review*, 9(2):246–275, 1991.

[20] Phillip D Brooker. *Programming with Python for Social Scientists*. SAGE Publications Limited, 2019.