Teaching Data Science to School Kids

Shashank Srikant and **Varun Aggarwal**
Aspiring Minds
{shashank.srikant, varun}@aspiringminds.com

Abstract

Data-driven decision making is fast becoming a necessary skill in jobs across the board. The industry today uses analytics and machine learning to get useful insights from a wealth of digital information in order to make decisions. With data science becoming an important skill needed in varying degrees of complexity for the workforce of the near future, we felt the need to expose school-goers to its power through a hands-on exercise. We organized a data science camp for kids in grades 5 through 9. Our aim was to expose them to the full cycle of a typical supervised learning approach - data collection, data entry, data visualization, feature engineering, model building, model testing and data permissions. We discuss herein the design choices made while developing the dataset, the method and the pedagogy for the camp. These choices aimed to maximize student engagement while ensuring minimal prerequisite knowledge. This was a challenging task given that we limited the pre-requisites for the kids to the knowledge of counting, addition, percentages and comparisons. By designing an exercise with the stated principles, we were able to provide to kids an exciting, hands-on introduction to data science, as confirmed by their experiences. To the best of the authors' knowledge, the camp was the first of its kind. Considering the successful execution of such a camp, we hope that educators across the world are encouraged to introduce data science in their respective curricula for high-schoolers and are able to build on the design principles laid out in this paper.

Data-driven decision making has become ubiquitous in businesses. One primary reason is that businesses have become 'digital' - customer acquisition, product/service development and delivery happens through the internet. A majority of our communication and social engagement also happens on the web. Unlike before, lots of data is now created, automatically recorded and is amenable to experiments. This makes data driven techniques efficient in providing solutions when compared to experts providing them (Lohr 2009; Manyika 2011; Kim and Begel 2010; Cárdenas-Navia and Fitzgerald 2015).

In the last decade, one major addition to the methods used in the industry to analyze information has been machine learning performing feature engineering to build predictive models (Breiman 2003). This is an addition to earlier methods of data analysis like studying group differences by an ANOVA or doing a linear regression. Feature engineering has become naturally important because of the need to derive insights from various types of unstructured data like text, voice, videos, etc. Modern analysis may take the form of both - supervised and unsupervised learning, the former being tractable for use by industries and often more useful for quick solutions. This new field of data analysis is loosely called data science.

In the near future, data science is likely to have a varying degree of influence on almost every available job. While some jobs will require the ability to record data in amenable formats and infer from predictive models, others will involve creating the right models and tools for analysis and insights (Diehl 2015; Chatfield et al. 2014; Swan and Brown 2008). For instance, a personal assistant will use an online tool to understand how his/her manager has spent her time in previous weeks to be able to predict meeting trends ahead in time and plan her schedule better. Such a reporting task will probably require recording data properly, cleaning it, bucketing it, being able to visualize it and deriving features for deeper insights. Likewise, a sales manager will be interested in answering a number of questions - what sales pitches work better, when to call a customer and what habits are common to productive sales executives. To learn all of this, the sales manager should first trust the fact that data-based methods can answer such questions, then be able to collaborate with a data scientist to record data in a form convenient for analysis and finally know how to interpret results from such an analysis in the context of sales and marketing process knowledge, which only s/he has.

With such a growing demand for data science in various professions, we think that it will become a general employability skill which needs to be developed early on, much like basic computer skills. We decided to make a first attempt at teaching data science to school kids from the 5^{th} to 9^{th} grades. Without being too pedantic about what to teach, our goal was to give kids a hands-on experience of the full cycle of a supervised learning task - data collection, data entry, feature engineering, data visualization, model building, model testing and data permissions. We didn't want them to

spend time staring at video tutorials or looking at an instructor working her way through data. Our hypothesis is that a student learns best when she solves a problem herself using data science, is able to think of other problems she could similarly solve and question the ifs and whys of the different steps involved in the process so as to appreciate the nuances and the beauty the subject has to offer.

We conducted such a half-a-day long data science camp for 14 students on 13 June 2015 (with a follow up on 20 June 2015).[1] In our experience, we felt that the students were able to perform all the steps in the flow and understand the various insights the experiment had to offer. We have made available the design of the experiment, dataset, mentor and student experiences on a public website for the community to use and develop further.[2]

With the aim to keep the material accessible to students given their training in math, computers and cognitive development, we made a considerable number of decisions when designing this camp. This included the choice of the dataset, its construction, the choice of the modeling technique and the software platform used. We feel that one of the reasons for the successful execution of this camp was our ability to control the complexity of the experiment while giving the kids a rich experience of doing something new and exciting.

Specifically, the paper makes the following contributions

- We lay out a set of design principles to choose a problem statement and create a dataset to provide a hands-on experience to secondary school students of the whole flow of a supervised machine learning task.
- We also propose a simple interpretable supervised learning model, analogous to a game, which the kids can easily understand, build themselves and see in action.
- We demonstrate the design principles by actually constructing an exercise which requires the kids to know only counting, addition, percentages and comparisons. While ensuring that the cognitive load of the exercise is bounded, the kids get an exciting, rich experience of being able to visualize effects, identify features and predict an unknown.
- We believe that this is a first successful demonstration of teaching young kids data science and should be an encouragement to the community to investigate and build on this further.

This paper is organized as follows - §1 discusses the design decisions which were considered to ensure that the exercise was accessible to kids. §2 provides details of the predictor which the kids had to design. §3 describes the various steps involved in the entire exercise. §4 discusses observations from the camp and describes student experiences. Concluding remarks and future work is discussed in §5.

[1]Since then, another camp has been organized for 15 students on 9 September 2015

[2]The details of the webpage will be included in this draft after the blind-review process

1 Design Considerations

The following constraints were considered while designing our experiments. The aim was to maximize participation among the participants while ensuring that the material was readily comprehensible, intuitive and the pre-requisites for participating in the exercise was minimal. In the remainder of this paper, we refer to the intended audience for such a data science camp, students in the 5^{th} to 9^{th} grades, as our *target group* (TG). We also interchangeably refer to the dependent variables as output variables and independent variables as input variables respectively.

1.1 Problem Statement

Full Data Cycle The exercise should provide the TG with a hands-on exposure to the full cycle of a typical supervised learning approach - data collection, data entry, data visualization, feature engineering, model building, model testing and data permissions. Introducing unsupervised learning would be harder to relate to and understand and is hence avoided.

Relatable Dataset The TG should be able to relate to and be interested in the data set used. They should also find what the model may predict to be exciting. For instance, commodity prices and stock market information would be rich in data but wouldn't be appropriate. On the other hand, predicting the weather based on the clothes people wore could seem obvious, thereby underplaying the role of data science. In summary, use a dataset which is relatable, interesting and can lead to some exciting prediction.

Pre-built Datasets There are several datasets on the web, including those specifically built for educational purposes. Avoid using these curated datasets. The TG should be exposed to the process of data collection and entry, which is an important component requiring time and attention when solving real world problems. In the larger scheme of applying data science to their surroundings, such an exercise would be an essential first step. Moreover, being involved in collecting and entering data would also give them greater ownership around the exercise and enhance the activity element.

Binary Variables In order to get a sense of the input variables, the TG should be able to visualize them and infer whether they discriminate the output. Assuming the inputs to be continuous-valued, they should not be expected to plot a probability distribution to see, say, group differences. Plotting a scatter between the dependent and an independent variable to intuit a pattern, an increasing or decreasing relationship, in a typically noisy graph would also be hard. Discretizing the continuous inputs, a process which would require understanding thresholds and their effect, would make the overall exercise complex. Hence, to keep the exercise simple, use discrete binary values for the dependent and independent variables. This reduces to the traditional two-class classification problem with only binary features. Here, visualizations could be made by merely counting, say, how many times a feature was represented (was valued 1) in the

Problem statement	**Relatable dataset**	The dataset used in the exercise must be relatable to a high-schooler. Some bad examples - oil prices, stocks etc.
	Data collection	Participants need to collect and enter data. This would give them a sense of how data is retrieved and collected in real applications.
	Prediction	The final prediction ought to have an aha-moment. It should not be something obvious, such as predicting the weather by looking at ones' clothes.
Dataset	**Data-type**	It is best if the dataset has discrete-valued variables. It makes analyzing the data much easier.
	Independent variables	The dataset should contain at most 3-4 independent variables.
	Balanced dataset	To make feature engineering intuitive, ensure that each feature is represented equally, creating a fully balanced dataset.
	Unseen data	Let the unseen dataset again be balanced, with each feature being represented equally in it.
Model	**Model building**	Participants should be able to design a simple, interpretable model from the dataset.
	Arithmetic involved	The math involved to design such a model should be constrained to operations in counting, addition, percentages and simple if-then logic.
	Model design	The design of the model needs to be amenable to high schoolers. It needs to have intuitive properties like a point-based system which adds up when the most discriminating feature is present in a sample.
Platform	**Easy tech**	Microsoft's Excel should be the most the participants use. A full-fledged programming language like R or Python would be high on pre-requisites.
	Manual override	The exercises should be designed such that it does not rely on formulas alone. Filtering, counting, pivoting information in an Excel spreadsheet should be demonstrable by manually performing these actions as well.

Table 1: Design principles for a hands-on exercise in data science for kids

two classes of the output. It would reduce to a simple bar graph where one compared the heights of the different bars.

Additionally, have only 3-4 independent variables in the dataset. The TG should be able to clearly understand the relationship between the dependent and independent variables.

Balanced Dataset Ensure that the two categories in each independent variable are represented equally in the training set and also within each category of every the other independent variable in the dataset. This would mean having 2^n unique data-points for n independent variables. This has a couple of advantages. First, when the effect of each independent variable on the output is visualized, it would likely represent the actual trend and not a result of an over/under representation of categories of another variable.[3]. This increases the chance of succeeding in being able to demonstrate an intuitively correct result. Second, and more importantly, the TG has to just see how the ratio of the categories of an independent variable moved from the expected 1:1 (See §3 for more details). This reduces cognitive load without compromising on the accuracy of the experiment design. Additionally, the training and validation set sizes should be at least in the ratio of 2:1. This would mean having at least $(2^n \times 3)$ data points in total.

[3]This isn't entirely true since we are balancing known variables and categories only, whereas unknown variables or categories with varying distributions create an imbalance

1.2 Predictive Model

The TG should be able to design a predictive model themselves and also understand the intuition behind its working. Learning a linear logistic model manually was ruled out and so were any methods involving complicated means and standard deviation calculations. We were inspired by a naïve Bayes classifier to form the classifier which the TG would use. Abstractly, naïve Bayes assigns a probability to an input $\mathbf{x} = \{x_1, ..., x_n\}$ belonging to one of k classes (2 in our case), C_k as -

$$p(C_k|x_1, ..., x_n) \propto p(C_k) \prod_{i=1}^{n} p(x_i|C_k) \tag{1}$$

A classifier to select the best class for $\mathbf{x}$ would be -

$$\arg\max_{k=1,2} p(C_k|x_1, ..., x_n) \tag{2}$$

The realization of this equation requires counting, division, multiplication followed by an if-then type decision making to decide which class the data-point finally belongs to, thus being in line with our guiding principles. We simplified this further: .a. we did not consider the prior probabilities, since we had a balanced class .b. we reduced the conditional probability to a binary 1-0 value, 1 if it were more than 0.5, and 0 otherwise .c. we summed these values rather than multiplying them. This resulted in modifying (1) to -

$$p(C_k|x_1,...,x_n) = \sum_{i=1}^{n} d_i \text{ where } d_i = \begin{cases} 1, & \text{if } p(x_i|C_k) > \frac{1}{2} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Another way to think about this is an ensemble of single-node decision trees which vote together. A score of $+2$ is given if the feature exists and -2 if it doesn't. This is done for each feature and the final score is the sum of individual scores. The final classification is done on whether the sum is greater than 0 or not. With a plus/minus point system, we reduce the cognitive load of taking out averages and also doing the calculations for each class and then comparing. The TG has to only count, add and compare. This model is fairly easy to understand - games have a point-based system where an accumulated score decides win or loss. Here, points given to each class decide whether it finally wins.

1.3 Technology

Microsoft's Excel could be used as the technology platform for the exercise. The TG would have to familiarize themselves to be able to enter data into Excel, count manually, draw graphs and be able to write an IF-condition formula and copy it. The TG could additionally use filters to make the counting easy. Those not comfortable with using Excel commands can be guided to perform such tasks manually. In our experience, kids picked up MS-Excel fairly quickly and were able to actively learn these by example. We do think that building a better tool specifically for data science, something akin to Scratch for programming(Resnick 2009), will be very useful and a promising area of research.

1.4 Risks in the Design

- One risk with using an on-the-fly dataset is to not find any interesting trends or good predictions. This can be countered in a couple of ways. One way could be to design a dataset where there is already evidence of a relationship, the extreme example being collecting data for a known physical law. The other way could be to pilot the dataset beforehand with a few kids to see if something interesting comes out. We followed a mix of both these approaches.
- One objection could be that we have conveniently simplified our experimental setup to ensure that it is always successful. These techniques are not entirely correct and would fail for real world datasets. We are aware of this. Our aim was to show one successful application of data science to the kids through which we could instill in them that they could solve problems themselves and encourage them to explore more sophisticated techniques. We might be at the risk of being inaccurate in our process of simplifying things, but we think this is in line with *arundhati nyaya*, a useful eastern pedagogical tradition. It describes the act of teaching an approximately correct but palatable idea first before teaching a fully developed, correct and non-trivial idea.

2 Problem Statement

Considering the design choices mentioned, we decided that our TG should design a $friend\ predictor$. We arrived at this decision after eliminating a few other choices. For instance, we considered a movie preference predictor, but found that it did not work well in cases where kids had not seen a movie, creating buggy or missing data in the process.

The Friend Predictor Each TG member got a set of images containing kids' faces along with their names and hobbies. By looking at these images and the description provided therein, they had to decide whether they would befriend the kid shown in the image. This data would be used to design a predictive model to predict if a new kid was friendworthy.

We avoided showing the TG faces of real kids as that would have added variance which our small sample size wouldn't have done justice to. There were only four dimensions of variance in our sample set which our TG could implicitly consider in deciding whether they would befriend the kid in the image or not. These were - *gender*, *hobby*, *name* and *facial expression*. Hobbies could broadly be categorized into two - indoor and outdoor activities. Names in some geographies have evolved. There's a noticeable difference between old-sounding names from the new. We hoped the TG would take to these names differently. We made some of the boys and girls look gloomy and some cheerful. We wanted to see if the TG was affected by facial features.

This was posed as a problem in supervised learning, which had four independent variables and one dependent variable - the rating provided by the TG participants.

3 Exercise Workflow

In this section, we describe the sequence of operations which our TG went through as part of the data-exercise. We list out the nuances involved in each step.

Training and Validation Sets The data set was presented to the TG as flash-cards, each of which had an image, a description of names and hobbies and some space in its corner where it could be rated (see Figure 2). Considering the design principles discussed in §1, there were a total of $(2^4 \times 3)$ cards to be labeled. We also had placed 8 additional cards for the TG to practice. Each TG member saw the flash-cards in the same order. The cards were ordered as follows - the first eight cards were for practice. The next 32 was the training-set, which contained two sets of the 16 unique permutations of the 4 features. The last 16 was the hold-out set kept for validating the model.

Labeling the Data Each TG member was shown the set of 56 images to rate. They were asked to assign their ratings on a scale of 1-5, where 1 meant they would certainly not befriend the person in the image and 5 meant they were very sure of befriending. They were given 10-12 minutes to complete the annotation exercise. Once rated, the practice images were removed, the 32 images meant for training and the last 16 meant for hold-out were placed separate envelopes. In the remainder of this paper, we refer to a set of images belonging to a participant as $sample$. The envelope containing training images will be referred to as the *training sample* and the hold-out set as the *validation sample*.

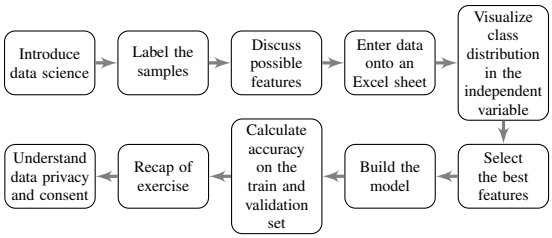

Figure 1: Workflow of our data science exercise

Introducing Features Through a lively discussion, the TG was encouraged to think out aloud possible features which could signal whether a person would befriend another. Amidst suggestions of befriending those who played the same video games and liking the same flavor of chocolates, the TG did conclude and saw reason in the four features we had set up and were ready to investigate the effect of each of these further.

Data Entry Each TG member was presented with the training sample of another TG member to analyze. Each of them had been provided with a computer containing a template Microsoft Excel workbook which had some prefilled information of the samples. The TG had to identify the features in each data-point of the training sample and enter them into the workbook. The order of entries presented in the workbook matched the order of the flash cards the TG had labeled - this made data entry convenient and confusion-free. A snapshot of the sheet presented to the TG is illustrated in Table 2.

Flashcard details			Features (To be filled by participants)			
Card #	**Name**	**Hobby**	**Gender**	**Hobby type**	**Name type**	**Rating**
1	Tanya	Badminton	*female*	*new name*	*outdoor*	*4*
2	Natasha	Painting	..	..	..	..

Table 2: Illustration of an Excel sheet used in the exercise

Visualization Once the information was entered in the sheets, we wanted the following questions to be answered

- Was a given sample "friendly"? Did the sample show an inclination towards befriending people or not.
- What features contributed in the friend-making process?

We considered visualizing the histogram of the ratings in the sample to help answer the first question (Figure 2). The TG manually counted the occurrences of each of the 1-5 ratings. They were then also shown how to apply filters to count the same. In order to readily see whether a sample had endorsed more friends than non-friends, we had to have the output in a boolean form. The TG took to it intuitively and could reason why any rating greater than 3 (or 4) could be binned as "will befriend" and the rest as "would not befriend". Once the output was binned into these two classes, the histogram was re-done.

The second question was answered by visualizing each feature's representation in the 'friend' and 'non-friend'

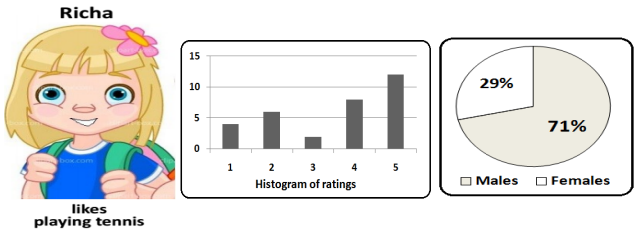

Figure 2: .a. A sample image which kids labeled .b. A histogram of labels .c. Pie-chart visualizing the number of males to females in those labeled as 'friends' in a sample

classes of the output. This also suggested which features were to be used in modeling the data.

Feature Selection By looking at pie-charts (Figure 2), the TG had to decide which features helped in discriminating between the 'friend' and 'non-friend' classes. For instance, if the ratio between the two categories of an independent variable in any one class of the output (we consistently considered the 'friend' class since the goal was to design a friend predictor) was, say, between $50{:}50$ to $60{:}40$, then the variable was considered to be unable to discriminate between the two classes. If the skew in the ratio was higher, it was considered as a discriminating feature and was used in the subsequent modeling process.

As a note, we would like to highlight here the advantage of having the same number of categories for each feature in the training sample. For instance, if we considered analyzing textitgender, the instructional statement to the TG would be - "Let us see what percent males are in the 'friend' class. 80% are males in this class as compared to 50% in the whole set; it can be thus inferred that this person prefers making male friends over females.". In the absence of a balanced number of categories, this would have been - "Let us see what percent males are in the 'friend' class and in the 'non friend' class. 80% are males in the former and 50% in the latter; it can thus be inferred that this person prefers making male friends over females.". The presence of equally represented categories eases the instructional overhead and the cognitive load of this exercise.

In order to verify how accurate the insights drawn from these visualizations and ratios were, members from the TG who had rated a sample were quizzed on whether these indeed were his/her tastes. This helped them get a sense of how the independent variables were indeed able to predict the dependent variable.

Model Building Once the TG had visualized which features were able to differentiate the dependent variable, they went ahead with the simplified model building exercise, detailed in §1. For each discriminating feature, the category which was represented higher in the ratio in the 'friend' class (referred to as 'dominated feature' in the subsequent sections) was counted in every point in the training sample. The occurrence of a dominated feature was given a score $+2$, its absence was given a -2. A threshold of > 0 was applied to

classify the data-point in the 'friend' class. This was implemented by using an IF-logic formula in Excel. The formula was demonstrated by the mentors for the first 2-3 data points of the training sample. The remainder was done by the TG.

Validation Once the TG had built their models, they tested how well it generalized on the hold-out set. The validation sample corresponding to each training sample was retrieved. An element of theatricality was added at this stage. The mentors built it up like an act of magic where they challenged the TG if the model could really predict. This created a sense of suspense and excitement in the group. The if-else deciding classifier was applied to the unseen points and the classification accuracy was noted down and tabulated.

Consent Once the TG noted the accuracy of their first classifiers, the entire exercise was summarized to ensure that they absorbed the key ideas. This was followed by an explanation of the importance of data privacy and being cognizant about the right to own the data they create. The importance of anonymizing information was explained. The TG then signed consent forms to allow their ratings and analysis be made public in an anonymized form so that the academic community could analyze it further.

4 Learning

We report here some experiences from the data science camp. We first see whether we succeeded with our dataset and models in making moderately accurate and generalizable models. This was a necessary condition for our success. We then discuss mentors' assessment of how the kids learnt and the difficulties they faced. We then describe the kids' experiences.

Results The exercise was successful in that we could see features which were able to differentiate the output classes. Of the 14 models built, 10 models had at least one discriminating feature and 3 models had two such features. The average train accuracy across the 14 models that were built was 78.2% and the average validation accuracy was 62.1%.

Mentors' observations It was essential that the kids were able to complete the exercise with the help of the mentors. In a focused discussion after the exercise, mentors agreed that kids were largely able to follow the key ideas of the exercise. They agreed that the kids could do the steps once they had demonstrated with an example. They reported that four kids had difficulty following the material. Among these four, three found dealing with percentages and ratios hard - no other math concept was identified which the kids found difficult. One kid just did not find the material and exercise interesting. One of them chose not to consent to share her data.

One may note that we did not use an objective assessment to test student learning at the end of the exercise - we did not intend do. In future work, we plan to give students a second exercise and see how well they do it independently.

Kids' experiences Kids found the problem statement to be fun and exciting. They were very comfortable in using MS Excel and picked it up rather quickly. They enjoyed making graphs and using different styles on them even though it wasn't required in the exercise. They each blogged what what they learnt, which is documented on our webpage. We describe here in brief their experiences and suggestions on scenarios where they thought they could use data science -

- "It was a very good experience and I loved it. I learned how to predict a stranger's choices just based on his or her data. It was very good."
- "I really enjoyed attending this workshop. I learned about data science and working with MS exel. I learned that foo likes outdoor games more than indoor" [sic.]
- "Exams. I would take my exam results, from the report card of every year. And then I will make it on excel and then I will remember the grades and the one I get more grades I will take a gift" [sic.]
- "I would like to predict my spending."

5 Conclusion and Future Work

The paper describes a half-a-day long data science camp that was organized for kids from grades 5 through 9. The camp was designed to expose them to the full cycle of a typical supervised learning approach. During the course of this camp, kids experienced how they could use data science to successfully solve a relevant problem, and in doing so, also appreciate the power and the applicability of such a technique. We share a set of design principles to help choose a problem statement, create a dataset and create a modeling technique such that it maximizes participation while ensuring minimal pre-requisites from the kids. The exercise was fairly successful, as signaled by both, the mentors' and kids' feedback, and also led to some learning for further design. In future, we plan to use more methods to gauge student learning.

We believe that our work presents a good starting point for educators to explore this further. We strongly believe that data science needs to be inculcated in school curricula and we see this as a resource which educators and curricula designers could take inspiration from. From a research position, a Scratch equivalent for data science poses to be a promising area of research at the intersection of HCI, programming languages and machine learning. Finally, we would also like to explore how we could scale the organization of these camps so that we could maximize its outreach to school students.

References

Breiman, L. 2003. Statistical modeling: The two cultures. *Quality control and applied statistics* 48(1):81–82.

Cárdenas-Navia, I., and Fitzgerald, B. K. 2015. The broad application of data science and analytics: Essential tools for the liberal arts graduate. *Change: The Magazine of Higher Learning* 47(4):25–32.

Chatfield, A. T.; Shlemoon, V. N.; Redublado, W.; and Rahman, F. 2014. Data scientists as game changers in big data environments. ACIS.

Diehl, M. 2015. Why marketers really need to know about data science.

Kim, M., and Begel, T. Z. R. D. A. 2010. The emerging role of data scientists on software development teams.

Lohr, S. 2009. For today's graduate, just one word: Statistics.

Manyika, J. e. a. 2011. Big data: The next frontier for innovation, competition, and productivity.

Resnick, M. e. a. 2009. Scratch: programming for all. *Communications of the ACM* 52(11):60–67.

Swan, A., and Brown, S. 2008. The skills, role and career structure of data scientists and curators: An assessment of current practice and future needs. report to the jisc. *Truro: Key Perspectives Ltd. Retrieved* 26(2):2013.