Embedding Data Skills in Research Methods Education:

Preparing Students for Reproducible Research

Phil McAleer[1], Niamh Stack[2], Heather Cleland Woods[1], Lisa DeBruine[1], Helena Paterson[1], Emily Nordmann[1], Carolina E. Kuepper-Tetzel[1], and Dale J. Barr[1]

1. School of Psychology and Neuroscience, University of Glasgow, Glasgow, United Kingdom
2. Department of Psychology, Mary Immaculate College, Limerick, Ireland

Author Note:

Data and scripts in R and Python for the analyses presented in this article can be found in our project repository at https://osf.io/7fs2b/. Correspondence concerning this article should be addressed to: Dale J. Barr, School of Psychology and Neuroscience, University of Glasgow, 62 Hillhead Street, Glasgow G12 8AD, United Kingdom. Email: dale.barr@glasgow.ac.uk.

Abstract

Many initiatives to improve reproducibility incentivise replication and encourage greater transparency without directly addressing the underlying skills needed for transparent and reproducible data preparation and analysis. In this paper, we argue that training in data processing and transformation should be embedded in field-specific research methods curricula. Promoting reproducibility and open science requires not only teaching relevant values and practices, but also providing the skills needed for reproducible data analysis. Improving students' data skills will also enhance their employability within and beyond the academic context. To demonstrate the necessity of these skills, we walk through the analysis of realistic data from a classic paradigm in experimental psychology that is often used in teaching: the Stroop Interference Task. When starting from realistic raw data, nearly 80% of the data analytic effort for this task involves skills not commonly taught—namely, importing, manipulating, and transforming tabular data. Data processing and transformation is a large and inescapable part of data analysis, and so education should strive to make the work associated with it as efficient, transparent, and reproducible as possible. We conclude by considering the challenges of embedding computational data skills training in undergraduate programmes and offer some solutions.

Embedding Data Skills in Research Methods Education: Preparing Students for Reproducible Research

Modern research faces major challenges from two sides: one technological, the other, epistemological. On the technological side, advances in information technology have made it easier than ever to collect, store, and manipulate data, resulting in datasets whose volume and complexity pose challenges to traditional analysis workflows. On the epistemological side, researchers face rising concerns about the reproducibility of published research. According to an online survey, about half of all researchers perceive a crisis of reproducibility in science (Baker, 2016). Across many fields including psychology (Hardwicke et al., 2018), political science (Eubank, 2016) and economics (Chang & Li, 2017), researchers frequently report difficulties computationally reproducing published findings from underlying data. Attempts to replicate published findings in new samples also suggest widespread problems in research practice: for example, in a large replication study by the (Open Science Collaboration, 2015), nearly two-thirds of experiments published within the same year in three of psychology's top journals failed to replicate. In what follows, we argue that success in addressing this twin set of challenges will be limited until they are addressed at the root; namely, by a curriculum change to research methods training.

The traditional data analysis workflows that students are typically taught within their academic fields have not kept pace with the increasing volume and complexity of datasets. The skills students need for working with real data are often only taught in specialised “Data Science” programmes, a discipline which has recently emerged out of ideas that long existed at the margins of traditional statistics (Donoho, 2017). Data scientists have developed powerful tools and guidelines for structuring and transforming data. Recognizing the value of these skills, undergraduate programmes may require students to learn coding and data science skills by taking classes offered by such programmes, outwith their specific field of study. In contrast, we argue that these skills should be directly embedded in field-specific research methods training, not only because such skills are fundamental to all modern empirical research, but also because fields differ in what aspects of these skills should be emphasized, depending on whether students will be expected to run online surveys, analyse genomic samples, wrangle gigabytes-worth of brain imaging data, or extract insights from large text corpora or laboratory experiments. Moreover, students are likely to learn better when given data that is relevant to their field of study (van Gog et al., 2019).

Embedding training in data skills within field-specific undergraduate research methods education also seems like a sensible strategy for improving long-term reproducibility. To date, however, initiatives to improve reproducibility tend to focus on restructuring incentives and practices for practicing researchers (Munafò et al., 2017; Nosek et al., 2012), and less emphasis has been placed on reforming the knowledge and skills acquired by students starting out on their career paths (but see Azevedo et al., 2019; Button et al., 2020). Researchers have been encouraged to adopt higher standards for transparency when publishing and reviewing papers (Simmons et al., 2016), pre-register study protocols and hypotheses (Wagenmakers et al., 2012), publicly deposit the data and code that

underlie published findings (Rouder, 2016; Wicherts & Bakker, 2012), and pool resources across labs by forming large research consortia (Moshontz et al., 2018). But until the skills that are needed to analyze data in a fully reproducible way become fully embedded from the beginning of students' careers, these initiatives, while laudable, may yield research artefacts that only superficially comply with open science guidelines. For instance, we may end up with published articles containing more detailed descriptions of methods that are neither more accurate nor more verifiable than previously, pre-registration protocols that cannot be fully audited because data processing actions were not logged, raw data without any computational roadmap for generating the findings reported in the paper, and research consortia datasets so large that few researchers are equipped to validate analyses performed on them.

The solution we propose involves moving away from the manual "point-and-click" workflows that have been popular in psychology and other academic fields. Such workflows are likely to impair reproducibility and transparency for the sake of perceived convenience. The plethora of point-and-click actions undertaken to perform a given analysis are often left unrecorded, leaving no clear record of how to derive the findings from the raw data. While many statistical software packages such as SPSS offer options to save the commands underlying specific analysis, it is often the case that the data input to SPSS has already been prepared using spreadsheet software such as Microsoft Excel. It is rare for researchers to manually log each transformation with enough detail for complete computational reproducibility (e.g., "step 273 of 521: filled down from cell G2 to G27 in ExperimentData.xlsx"). Moreover, the common use of multiple distinct software packages for different stages of analysis—Excel for pre-processing and visualisation, SPSS for analysis, Word for writing the report—can increase the risks of outputs becoming out of sync, such as when a researcher corrects an error in pre-processing but then analyses the old version of the dataset in SPSS. The prevalence of point-and-click workflows means that the descriptions of data analysis procedures in most published papers are largely post-hoc verbal reconstructions from memory. As such, they are likely to contain omissions, ambiguities, and biases (Schacter, 1999).

Researchers need data analysis workflows that are simultaneously reproducible, transparent, and efficient. We believe that the best approach to ensuring future researchers produce reproducible research is for undergraduate students to be taught and trained in data skills, through writing analysis scripts using code and through field specific research methods programmes infused with ideas from Data Science. Unlike point-and-click analyses, scripted analyses are self-documenting. Writing each data processing step down as a function call in a script removes the additional burden to separately log each processing step: the script contains the logic for the complete analysis. Furthermore, it is necessary to train students in skills related to "data wrangling," as this best prepares them for the challenges of real data. Currently, it is uncommon to find emphasis on these topics in undergraduate research methods classes (Zečević et al., 2021), although there are exceptions (e.g., Auker & Barthelmess, 2020; Baumer et al., 2014; PsyTeachR Team, 2022; Toelch & Ostwald, 2018). For convenience we will refer to this missing set of scripting, visualisation and data wrangling skills as "computational data skills." We acknowledge that coding, visualisation and data wrangling are not the only computational data skills,

however, learning these three computational skills acts as an extensive introduction to teaching reproducible analysis. Graduates may not just need these computational data skills to work competently in their fields, but they need them to succeed in an economic marketplace where data science skills are increasingly in demand (Bradford, 2018). There are also societal benefits to having a technologically informed and critically empowered citizenry. In the United States, science authorities recommend that institutes should prepare all their students for the new data-driven era with appropriate data science skills and *data acumen*, an umbrella term stretching from data management and curation to ethics, as well as data wrangling, modelling and reproducibility (National Academies of Sciences, Engineering, and Medicine Committee on Envisioning the Data Science Discipline, 2018). In the United Kingdom, the president of the Royal Society has recently likened a society lacking in data literacy to the mass illiteracy of the past, emphasizing how the digital era has changed the nature of work (Smith, 2022), requiring educational reform to address the skills gap (see also a report from the National Foundation for Educational Research, Taylor et al., 2022). Research methods curricula are a good place to address these needs by introducing changes that help develop problem solving, numerical literacy, and critical thinking, as these are all skills that students need to independently carry out a piece of empirical research.

When evaluating a curriculum, a useful exercise is to step back and ask, *What is something observable that you think students in your field ought to be able to do when they graduate, and are you adequately preparing them to do this?* (Nolan & Temple Lang, 2010; Peck and Chance, 2007). As academic psychologists in the United Kingdom, we adhere to the Quality Assurance Agency for Higher Education's Benchmark Statement (2019), according to which all final year students must be able to "carry out an extensive piece of empirical research that requires them individually to demonstrate a range of research skills, including planning, considering and resolving ethical issues, analysis and dissemination of findings" (p. 6). These benchmark statements inform the British Psychological Society Standards for Accreditation (2019) which all accredited undergraduate psychology programmes in the UK must adhere to. In many programmes, it is only in the course of conducting this final research project that students have their first contact with realistic datasets, typically from online surveys or laboratory experiments, as often idealised data is employed in classroom exercises. In this paper, we show how one of the simplest and most well-known experiments in our field of psychology generates data processing challenges for which students who receive the traditional point-and-click training are likely to be unprepared at dissertation point or post graduation. We then discuss how we can redesign curricula to meet these challenges. We start by presenting the data in an *idealised* way, which is how students will often receive data for analysis in class. Idealised data omits all initial data processing steps and presents data in a format that is convenient for statistical analysis, but that bears little resemblance to raw data coming straight from a study or experiment. We then provide a full, *realistic* walkthrough that includes all data processing steps from data collection to final analysis. We suggest that it is only through repeated practice with realistic data that students can fully develop their computational data skills. To show the applicability of these skills beyond experimental data, in the supplementary materials, we consider a scenario where the goal is just to calculate a score from survey data, and there is no sample-to-population inferential analysis.

Task analysis: Analysing Stroop data

One of the oldest and best-known paradigms in experimental psychology is the Stroop task (Stroop, 1935). In a standard version of the task, participants are given a list of words and must name aloud the colour in which each word is displayed. What makes this task psychologically interesting is that the words themselves can be colour words (RED, BLUE, GREEN, etc.) whose meanings may match or mismatch the display colour. For example, the word RED printed in red would require the response "red", while seeing the word RED printed in green would require a response of "green". The case where the ink matches the word is the **congruent** condition, and the case where it mismatches is the **incongruent** condition. Participants are shown one word at a time, and the experimenter measures the amount of time it takes them to name the ink colour, starting from the onset of the presentation of the word (this quantity is known as their "response time"). Each stimulus word paired with the spoken response is referred to as a single **trial** in the experiment. A standard finding is that participants are slower and less accurate to name the word's display colour when it mismatches the colour the word denotes (Dalrymple-Alford & Budayr, 1966), yielding the highly replicable "Stroop interference effect." Because nothing in the task requires identifying the words or accessing their meanings, this interference suggests that the reading of written words may occur somewhat automatically. The simplicity of the Stroop experiment and the high replicability of the effect makes it a popular choice for use in teaching.

Let's imagine that students in a statistics and research methods course are given data from a variation on the basic task. The goal of this imaginary experiment is to test whether people who speak English as a second language are more or less susceptible to Stroop interference than those who speak English as a first language. In the following two sections, we will consider two very different representations of data from such a hypothetical experiment: an idealised version where the data have already been heavily cleaned and pre-processed into participant means and a realistic version that more accurately reflects the complex and messy nature of the raw data as it comes out of the experiment. Although the data in the examples below is artificial, we have simulated it to capture many of the properties of real data, including the need to combine different data sources and discover and repair inaccurate values. To simplify our presentation, we discuss the analysis steps in a general manner rather than including any code. The simulated data and analysis scripts in R and Python can be found in the project repository at https://osf.io/7fs2b/.

Stroop Analysis: Idealised Data

Real data rarely first appears in a format that is convenient for analysis, often requiring extensive transformations before statistical procedures can be applied. During statistical training, instructors usually omit any pre-processing stages, presenting the data in an idealised format tailored to the demands of statistical software, such as the data shown in Table 1. Let us imagine we give this dataset to students along with the instruction:

> Table 1 has participant means for response time in milliseconds (ms) in the congruent and incongruent conditions. Calculate a Stroop effect for each participant, and then analyse the data using a one-sample *t*-test to test the overall

> Stroop effect (ignoring group), followed by an independent-samples t-test to compare the effect across English language groups. Perform both as two-tailed tests with $\alpha = .05$ and report your results in APA format. In addition to your test results, report means and standard deviations, measures of effect size, and 95% confidence intervals.

Table 1. Idealised Stroop data, showing participant response times (ms) means for the first three and last three participants.

eng_lang	congruent	incongruent
first	587	710
first	436	571
first	601	803
...	...	...
second	676	903
second	461	492
second	467	622

The data in Table 1 is in wide format—a format familiar to users of software such as Excel or SPSS—where each row represents data from a single participant and the observations across the conditions of congruent and incongruent are represented across columns. The dataset is readymade for analysis using conventional software; indeed, the structure of the data even suggests the type of analysis that is expected, making it easy to see how to calculate the Stroop effect for each participant by taking the difference between the values stored in the congruent and incongruent columns.

Box 1 presents a sample Results section in APA format that might be prepared from the statistical output. Writing such a section requires calculating 23 separate data-dependent values (textual as well as numeric), all of which are underlined.

Box 1. Sample Results for Idealised Data.

We tested 40 participants on a five-colour Stroop task.

On average, speakers responded 184 milliseconds (SD = 69) faster in the congruent than in the incongruent condition, $t(39) = 16.95$, $p < .001$, $d = 2.68$, 95% CI [162, 206].

Participants who spoke English as a first language (N = 22) showed an average Stroop effect of 193 ms (SD = 71), compared to an average of 174 ms (SD = 67), for participants who spoke English as a second language (N = 18). According to a two-tailed independent-samples t-test with $\alpha = .05$, the group difference was not statistically significant, $t(38) = .85$, $p = .401$, $d = .27$, 95% CI [−26, 63].

There are essentially six steps required to compute these values (code to do so in R is provided in the project repository):

1. Count the number of participants overall and in each group;
2. Calculate a Stroop effect for each individual participant;
3. Calculate the overall mean and SD for the new Stroop effect variable;
4. Conduct a one-sample t-test on the Stroop effect variable, with effect size and confidence intervals;
5. Calculate the mean Stroop effect with SD for each language group;
6. Conduct an independent-samples t-test to compare the Stroop effect by language group, with effect size and confidence intervals.

However, these steps still comprise a small minority of the total effort that would be needed if students were given a more realistic starting point for their analysis: the raw data.

Stroop Analysis: Realistic Data

Transforming raw data into a format suitable for analysis typically involves many more processing steps and analytic decisions than would be needed for calculating descriptive and inferential statistics from idealised data. To see this, let us now turn to the raw data used to generate the participant means in Table 1.

The data that we present below has all the components needed to calculate the participant means shown in Table 1, but as we will see, the data processing steps needed to do so are neither self-evident nor trivial, and so we will thoroughly explain each step as we progress. Those familiar with R or Python code can view the corresponding scripts in the project repository (available in the OSF repository as a plain R script, RMarkdown notebook with R code, or Quarto notebook Python code).

Table 2. Demographic data

id	age	eng_lang
1	226	first
2	21	second
3	21	first
4	21	second
5	22	
6	24	fist
…	…	…
42	23	second
43	18	first
44	24	second

When participants arrived in the lab, the experimenter collected demographic information (participant age and whether they spoke English as a first or second language). To ensure anonymity, each participant was assigned a unique integer number (ID). These data were typed into a spreadsheet represented in Table 2. For their language background, the experimenter entered one of two values for the variable *eng_lang*, either *first* or *second*– at least, that is what the experimenter intended to do, but they didn't carry out their task flawlessly. First, there are 44 rows in Table 2, four more than in the table of participant means (Table 1). Data from these additional participants cannot be used because, for some reason, the value for *eng_lang* was not recorded. Additionally, for participant 6 we can see that the experimenter has typed *fist* instead of *first*. We would need to check for further typos and inconsistencies (e.g., other variations in spelling or in capitalisation) before we can safely use the data. Typos are also likely in manually entered numeric data. For instance, one of the ages has been erroneously entered as 226. To detect such anomalies, it is important to check data distributions for any manually entered numeric values.

The participant means in Table 1 were averages taken over a series of individual trials in each condition. Specifically, in this experiment there were 50 trials for each participant, 25 where the text and colour were *congruent* and 25 where they were *incongruent*. As is typically the case, stimulus presentation was controlled by computer software which generated a text file containing a stream of timestamped events such as those shown for participant 12 in Table 3. We have 44 of these files, one for each participant, which must be imported and combined into one larger table.

Table 3. Timestamps for subject 12.

trial	timestamp	event	data
1	175592	DISPLAY_ON	GREEN-green.png
1	176159	VOICE_KEY	
2	178485	DISPLAY_ON	BROWN-brown.png
2	179142	VOICE_KEY	

trial	timestamp	event	data
3	181146	DISPLAY_ON	BLUE-red.png
3	182165	VOICE_KEY	
4	184646	DISPLAY_ON	PURPLE-red.png
4	185706	VOICE_KEY	
5	187558	DISPLAY_ON	GREEN-green.png
5	187963	VOICE_KEY	
6	190339	DISPLAY_ON	GREEN-brown.png
7	192874	DISPLAY_ON	BLUE-red.png
7	193606	VOICE_KEY	
…	…	…	…
48	307302	DISPLAY_ON	RED-purple.png
48	308314	VOICE_KEY	
49	310420	DISPLAY_ON	PURPLE-purple.png
49	310813	VOICE_KEY	
50	312647	DISPLAY_ON	BROWN-brown.png
50	313049	VOICE_KEY	

The output in Table 3 contains information about the trial number (1 to 50), event timestamps, and associated data. Importantly, it does not contain any explicit information about what condition (congruent or incongruent) a trial was in, nor about the response time for that trial. We have the information we need but converting this information into actual data values that we can use in an analysis will require some data processing steps.

First, we can get information about each trial's condition from the stimulus filename, which appears in the *data* field for *DISPLAY_ON* events. On each trial of the experiment, an image file in Portable Network Graphics (PNG) format would be displayed, centered on the screen. Each file contained image data of a word in a particular font colour, with the filename containing metadata indicating the identity of the word being displayed (in capital letters) and the display colour (in lowercase letters). For instance, a file named *RED-green.png* would display the word RED in a green colour. Presentation of the image would trigger a *DISPLAY_ON* event, with the timing of the event on the computer's internal millisecond clock stored in the *timestamp* field. To determine what condition the trial was in, it is necessary to parse out the two colour values from the filename and compare them, such that, for instance, *GREEN-green.png* would be marked as congruent and *BLUE-red.png* as incongruent. A table containing this information appears as Table 4.

Table 4. Intermediate table showing trial conditions for Participant 12.

trial	data	stimword	inkcolour	condition
1	GREEN-green.png	GREEN	green	congruent
2	BROWN-brown.png	BROWN	brown	congruent

trial	data	stimword	inkcolour	condition
3	BLUE-red.png	BLUE	red	incongruent
4	PURPLE-red.png	PURPLE	red	incongruent
5	GREEN-green.png	GREEN	green	congruent
6	GREEN-brown.png	GREEN	brown	incongruent
…	…	…	…	…
48	RED-purple.png	RED	purple	incongruent
49	PURPLE-purple.png	PURPLE	purple	congruent
50	BROWN-brown.png	BROWN	brown	congruent

Response time was measured by a 'voice key' algorithm that detects the onset of speech. After the stimulus appeared, the participant would name the display colour aloud into a microphone connected to the computer. The voice key algorithm triggered a *VOICE_KEY* event at the first moment when a vocal response was detected. Our dependent variable, response time, would be calculated as the latency between the two timestamps for each trial. But as can be seen for trial 6 in Table 3, the algorithm could sometimes fail, in which case the *VOICE_KEY* event would be missing, and no response time could be calculated.

Calculating response time from these data requires restructuring it from long to wide format (also known as "pivoting" the data), such that each trial is represented in a single row, with *DISPLAY_ON* and *VOICE_KEY* now appearing as variables whose values are the associated timestamps for the corresponding trial (Table 5). Response time could then be easily calculated by subtracting the *DISPLAY_ON* timestamp from the *VOICE_KEY* timestamp for that row. Note that missing values now appear as *NA* ("Not Available").

Table 5. Transformed trial data with calculation of RT (VOICE_KEY - DISPLAY_ON).

trial	DISPLAY_ON	VOICE_KEY	rt
1	175592	176159	567
2	178485	179142	657
3	181146	182165	1019
4	184646	185706	1060
5	187558	187963	405
6	190339	NA	NA
…	…	…	…
48	307302	308314	1012
49	310420	310813	393
50	312647	313049	402

We have computed the trial condition as well as the response time, but we are not yet ready to calculate participant means for each condition until we deal with inaccurate trials. We would expect that on a minority of trials, participants would accidentally name the

word instead of the display colour. Because the response distributions for accurate and inaccurate trials would differ, it would be advantageous to identify inaccurate trials and exclude them from the analysis.

Determining accuracy for each trial requires identifying each participant's vocal response on each trial. During the experiment, the experimenter transcribed these vocal responses in real time into a spreadsheet file (see Table 6). Since the vocal responses were typed manually and in real time into a spreadsheet as the experiment progressed, we should expect typos, and indeed, among the 2,200 values there are not only the five correct values (red, green, purple, blue, and brown) but also 80 erroneously typed variants such as "bleu", "borwn", "geren", "pruple", and "rde". Typos such as these would need to be cleaned up before the values could be used to determine accuracy.

Table 6. Vocal responses transcribed by the experimenter.

id	trial	response
1	1	red
1	2	green
1	3	red
…	…	…
44	38	purlpe
44	39	brown
44	40	red
44	41	green
44	42	blue
44	43	blue
44	44	purple
44	45	blue
44	46	red
44	47	brown
44	48	blue
44	49	green
44	50	red

The need to compare the cleaned values in the *response* variable of Table 6 to the values of the variable *inkcolour* in Table 4 poses a problem: How do we perform computations that involve variables from separate tables? Not only do we need to solve this problem to compute accuracy, but also to calculate participant means, because doing so requires bringing together variables that are currently scattered across various tables: *eng_lang* in Table 2, *condition* in Table 4, and *rt* in Table 5. Combining information from multiple

distinct tables is a problem that is frequently encountered with real data. The manual solution of copying and pasting data is as tedious as it is error prone[1]. Fortunately, there is an easy and powerful computational solution: using a single database-style "join" function, where values from a pair of tables are merged based on common values in one or more "key" variables.

Table 7. Trial, transcript, and demographic data combined in a single table.

id	eng_lang	trial	stimword	inkcolour	condition	response	is_accurate	rt
1	first	1	RED	red	congruent	red	TRUE	757
1	first	2	GREEN	red	incongruent	green	FALSE	754
1	first	3	RED	green	incongruent	red	FALSE	798
…	…	…	…	…	…	…	…	…
12	first	1	GREEN	green	congruent	green	TRUE	567
12	first	2	BROWN	brown	congruent	brown	TRUE	657
12	first	3	BLUE	red	incongruent	red	TRUE	1019
12	first	4	PURPLE	red	incongruent	red	TRUE	1060
12	first	5	GREEN	green	congruent	green	TRUE	405
12	first	6	GREEN	brown	incongruent	brown	TRUE	NA
…	…	…	…	…	…	…	…	…
44	second	48	BROWN	blue	incongruent	blue	TRUE	401
44	second	49	PURPLE	green	incongruent	green	TRUE	593
44	second	50	PURPLE	red	incongruent	red	TRUE	498

To compare the values of *response* and *inkcolour*, we would need to join Table 6 to Table 4. This would require matching rows on the values of *id* and *trial*; however, Table 4 lacks the *id* variable. For the trial data from which Table 4 was derived, the *id* value was given in the filename (e.g., *S12.csv*). During file import, we would need to parse out the value 12 from the filename and add this variable to the trial data. We could then easily join the data from these two tables, and calculate the variable *is_accurate* by comparing *response* to *inkcolour*. This join will also bring *condition* into the resulting table. Following the same logic, we then bring in the *rt* variable by joining the result to Table 5 on the key variables of *id* and *trial*. Finally, we add *eng_lang* by joining Table 2 to the latter result, yielding a table that has all the variables we need to compute subject means in one place (Table 7).

[1] Excel's VLOOKUP function is one solution to joining data from different worksheets, but has a number of limitations as compared to SQL-style "join" functions available in R or Python; see https://www.quora.com/What-are-the-limitations-of-VLOOKUP.

What remains is to calculate means for each participant in each condition, and then calculate the Stroop effect by subtracting each participant's congruent mean from their incongruent mean. This requires grouping the data by the unique `id` values, calculating means, and then pivoting the table from long to wide in order to calculate the difference. This final series of steps would yield Table 1, the "idealised" data that served as our starting point in the previous section. The rest of our analysis would proceed as it did in that section.

But we are not done until we have written our results up, incorporating all the statistical quantities that we computed. The sample Results section in Box 2 is much closer to the kind of results section one would find in an actual manuscript than Box 1, in that it describes and justifies participant and trial exclusions, in addition to reporting descriptive and inferential statistics. In Box 2 we underlined the 30 values (numbers or words) in this section that are determined by the data, seven more than were computed for the idealised data. All of these additional values appear in the second paragraph of the results, which describes exclusions that took place both at the participant level (because the experimenter forgot to record the participant's English language background) or at the trial level (because participants erroneously named the word rather than the colour or because the voice key algorithm failed). The remaining paragraphs are identical to those in Box 1.

When using point-and-click software, transcribing values into a Results section must be done manually, and is therefore subject to transcription errors. For instance, it is estimated that about half of all published papers in psychology contain at least one inconsistent statistical value, such as a p-value that does not correspond to the reported t-value and degrees of freedom (Nuijten et al., 2016). A more reliable solution that we can strive to include in our curriculum is to use a 'literate programming' (Knuth, 1984) or 'notebook' approach, where code and plain text are combined in a source document, which is then compiled into a report where values from the analysis are automatically updated and formatted appropriately. For instance, this can be accomplished by embedding the analysis into a document written in R Markdown (as we have done for this paper) or by using a Quarto or Jupyter notebook. The examples in the supplementary materials show how this can be done in more detail. Writing a fully dynamic manuscript that is then compiled to HTML or LaTeX can be a high bar for beginners, and so a more accessible but less optimal solution is to copy the generated values into the manuscript in one go. This is still better than having to transcribe or copy-paste each value independently, and indeed, there is evidence that students prefer this method (Baumer et al., 2014).

Box 2. Sample Results for Realistic Data.

We tested 44 participants on a five-colour Stroop task.

We had to remove data from four participants whose English language background was not properly recorded by the experimenter. From the full set of 2,200 trials recorded for the remaining participants, we removed 45 trials (2.0%) where participants produced the wrong answer and 19 (0.9%) further trials that could not be analysed because of voice key failure. This left 2,136 trials for analysis. For each participant, we calculated the mean response time in the congruent and incongruent condition.

On average, speakers responded 184 milliseconds (SD = 69) faster in the congruent than in the incongruent condition, $t(39) = 16.95$, $p < .001$, $d = 2.68$, 95% CI [162, 206].

Participants who spoke English as a first language (N = 22) showed an average Stroop effect of 193 ms (SD = 71), compared to an average of 174 ms (SD = 67), for participants who spoke English as a second language (N = 18). According to a two-tailed independent-samples t-test with $\alpha = .05$, the group difference was not statistically significant, $t(38) = .85$, $p = .401$, $d = .27$, 95% CI [−26, 63].

Discussion

In research methods classes as well as in typical statistics textbooks, datasets are usually presented to students in an "analysis ready" format, such as the idealised data presented above in Table 1. While there may be short-term pedagogical advantages to doing so, the routine use of such idealised data is likely to deprive students of important opportunities to develop and refine a range of computational data skills. As we have demonstrated, pre-processing realistic data into an "analysis ready" format involves many steps, even in the presumably simple case of the Stroop Interference Task. The number and complexity of these steps makes data pre-processing errors likely, and they will be difficult to detect if students are using a traditional "point-and-click" workflow. By writing each action down as a function call within a script, all analysis decisions are made explicit, including those involved in data pre-processing, increasing transparency and computational reproducibility. Moreover, a manual point-and-click workflow is inefficient and even impractical for large datasets. To develop their competence and confidence, students require training in computational data skills and repeated opportunities to use them. In this section, we characterise these skills more precisely based on the task analysis we presented above. In addition, we offer suggestions for revising research methods curricula.

Our script to process the realistic raw data required 78 function calls, as compared to just 14 function calls for the idealised data. If we consider a function call to be a single action, and we consider our Stroop analysis as representative of a typical psychology experiment, then it follows that students who are only trained on idealised data sets are missing out on

about 80% of what they need to know to work efficiently, accurately, and reproducibly with data.

Much of the analysis involved transforming data stored in tables, or *tabular* data. Students could benefit by learning principles about how to best structure tabular data to enable efficient processing, including guidelines for data entry and storage (Broman & Woo, 2018) as well as principles for organising data into a predictable "tidy" format so that a broad array of data processing functions can be easily chained together (Wickham, 2014). The raw Stroop data that we started with happened to be already structured in tidy format, but this is often not the case; for example, our analysis of personality questionnaire data in the supplementary materials starts out as "untidy" and requires transformation before analysis can proceed. Moving between data structures that are easy to read by humans and those that are useful for analysis or visualisation requires skills that only come with practice.

Box 3 offers a more detailed analysis of the types of functions that were needed (in R) for processing the realistic data, grouped into four categories. This makes evident that most of the work involved transforming data stored in tables. Of the 78 function calls, 32 (41%) involved data transformations (categories 2 and 3 of the functions in Box 3), whereas only nine (12%) were for mathematical or statistical calculation. In contrast, for the 14 total function calls in the script for the idealised data, only three (21%) involved the transformation of tabular data (categories 2 and 3 of the functions in Box 3), while seven (50%) involved mathematical or statistical calculations.

Box 3. Classes of functions needed for analysing realistic data in R.

1. **Import**

function	n*	description
`read_csv`	3	read CSV data into memory
`dir`	1	list all files in a directory

2. **Transform a Table**

function	n*	description
`mutate`	7	create new variables
`select`	6	include/exclude variables (columns)
`filter`	6	include/exclude observations (rows)
`group_by`	2	group observations into higher-level units
`summarise`	2	calculate summary statistics
`distinct`	2	enumerate distinct values or combinations of values
`pivot_wider`	2	convert long data into wide data
`separate`	1	split string values into new variables

3. **Combine Two Tables**

function	n*	description
`inner_join`	2	combine variables based on set intersection
`left_join`	1	add variables from 'right' table onto 'left' table
`semi_join`	1	keep observations in 'left' table that are in 'right' table

4. **Statistical and Mathematical Functions**

function	n*	description
`sd`	2	calculate standard deviation
`t.test`	2	run a t-test
`cohensD`	2	calculate Cohen's d
`mean`	1	calculate mean
`n`	1	count number of observations
`round`	1	round off a floating-point value

*the number of times the function appears in the script for realistic data

The idealised data not only hid all the transformations required to get the data into "analysis ready" format, but it also obscured the need to screen the data for problematic observations and deal with them appropriately. Checks for data quality and criteria for data exclusion are often part of the "hidden methodology" that can contribute to irreproducibility (Breznau et al., 2022). When using a manual point-and-click approach, exclusion operations can easily be forgotten and left undocumented, and recalled descriptions can differ from the actual process in ways that affect the outcome. For example, if you calculate the mean and standard deviation for excluding outliers before or after you exclude participants for failing to answer a critical demographic question, you can end up with different final datasets (Silberzahn et al., 2018). Starting from the raw data highlighted the imperfect nature of real data, and exclusion operations were self-documented in the code so that they could be accurately reported in the writeup.

Upskilling Staff

To transform a curriculum to include reproducible data skills, one needs a critical mass of academic staff who are proficient in the same language and analysis workflows. Staff who teach research methods are likely to need additional professional development to be comfortable teaching students how to code. There are several models for accomplishing this. If your department has enthusiastic and knowledgeable staff members, their

workloads could be adjusted to allow them to lead a series of tutorials for staff. External training programmes are another option, but these may be too generic for your specific needs. Staff from departments that have already made this transition may be willing to provide tutorials and financial arguments can be made that any upfront training costs will be offset by the eventual move away from expensive proprietary software licences to open-source software. The open research community has also created many learning resources for independent study. One discipline-agnostic resource is our Applied Data Skills book (Nordmann & DeBruine, 2022); staff who are comfortable with the exercises at the end of each chapter should be well-versed in the data wrangling skills required for undergraduate curriculums.

It is also important to recognise that training is not a one-off affair and that having a skill does not mean one is able to teach that skill. Upskilling should begin at least one academic year in advance of any curriculum change. It may also be wise to commit to yearly refresher training courses. Peer observation of teaching is an effective part of many teacher training programmes. From a workload perspective, one option to support upskilling we have found to be successful and sustainable is for more experienced and confident staff to offer peer observation with less confident staff assisting with training sessions. This enables trainee staff to practice helping learners with their code without having to lead the session.

Critical to the success of any curriculum reform is recognising that some staff may be apprehensive about a shift to teaching data skills programmatically and building in support mechanisms so that people can receive assistance in a supportive and respectful way. We urge those who are contemplating such a move to view it as an opportunity to build a supportive community that values reproducible research and computational data skills but also provides a non-judgemental and supportive environment for staff to upskill in. For instance, we have found it very effective to use messaging boards (e.g., on Slack or Microsoft Teams) to field questions from staff or students (each with their own channels), to share information about new software packages, or to share their excitement (or sometimes, frustration) about the new way of doing things.

Finally, upskilling requirements should factor into decision-making around which scripting language fits your needs. Consider the norms in your field, the skills already possessed by your staff, and the learning materials available. For example, R is the dominant scripting language in experimental and social psychology, while Matlab or Python is more common in psychophysics and neuroimaging. Open-source languages, such as R, Python, and Julia, have accessibility and cost advantages over proprietary languages, such as Matlab. Open-source alternatives to SPSS such as Jamovi and JASP may reduce upfront upskilling costs, but they are focussed on statistical tests and offer limited scripting options for data preparation.

Curriculum reform

A common concern we have encountered is that modern psychology undergraduate degrees do not have room in the curriculum to prioritise teaching reproducible data skills. A curriculum review is a useful place to start addressing this concern and provide guidance and a framework to support this process. With the data processing tasks described in this

paper in mind, determine which skills, if any, are being explicitly taught and also consider whether time spent on higher level statistical analysis may be more fruitfully spent ensuring basic data skills are developed. Less emphasis on inferential statistics and more on data literacy and visualisation will allow students to learn valuable skills and enhance reproducibility and transparency without falling afoul of accreditation guidelines for undergraduate programmes. Moreover, by emphasising computational and problem-solving skills that are broadly applicable to real data sets, we better serve the future employability of our students across a diverse range of post-graduation career paths.

Educators must accept that data processing and transformation is a large and unavoidable part of real data analysis. Accepting this means that we should strive to ensure that it is done in the most efficient, transparent, and reproducible manner possible. Sooner or later—whether during a final research project, or in one of the many jobs in the workplace that require analysing data and generating reports—our graduates will stumble upon real data. It is our responsibility as educators to ensure our students can meet its analytical challenges with confidence and efficiency rather than with frustration, self-doubt, and wasted effort. Current approaches leave significant gaps in their education. Developing students' data skills so that they can work more independently empowers students while also freeing up supervision time for higher-order discussions relating to analysis and interpretation.

To give an example of an undergraduate curriculum built around a foundation of computational data skills, in our four-year programme at the University of Glasgow, students learn how to wrangle and visualize data by developing scripts in R, using the RStudio Integrated Development Environment as an interface. As part of their assessment portfolio, students receive a plain-text R Markdown file with empty code chunks, where they have to fill in the correct code to perform the analysis. This enables them to become fluent in producing dynamic reports and allows assessment and feedback not only on their mastery of course content, but also on whether their analyses are reproducible. Whereas our past curriculum had them learning how to do a t-test in their very first lab, they now do not learn inferential tests until the start of their second year, after they have become proficient in importing, transforming, visualising, and summarising data. The second year provides a more traditional curriculum, except that most of the assignments involve realistic data and therefore allow students to practice and consolidate data wrangling skills. In the third year, which is also the final year of obligatory statistics training, students receive more advanced training in linear mixed-effects modelling. Despite spending more time on coding and data wrangling in our new curriculum, and introducing inferential statistics much later, we nonetheless found that students were able to "catch up" and attain similar levels proficiency in inferential statistics by the end of their third year. However, we acknowledge that the impact of these changes on students' statistical proficiency needs systematic empirical study. We also appreciate that our four-year degree programme allows us ample time to cover the relevant material, whereas psychology programmes are often run over three years. However, we have been successfully adapted shorter versions of our programme for one- and two-year conversion programmes, which shows the adaptability of the general approach.

A final consideration for curriculum reform is that the goal is to increase students' ability to handle and reason about data, so teaching these skills requires more than just changing software, for example swapping R for SPSS in existing exercises. For example, across the degree programme, our curriculum now includes registered reports, pre-registration, the use of secondary data that requires cleaning and wrangling, and learning about statistical models through simulation. One method of reducing the workload associated with such curriculum reform is to draw on existing open educational resources, such as the PsyTeachR series of online textbooks (https://psyteachr.github.io), which target a variety of audiences and skill levels. The textbooks are open-source and available under a CC-BY-SA license, which allows others to modify and re-use them in their own courses.

Conclusion

We hope to have made a convincing argument for teaching field-specific computational data skills as part of the undergraduate research methods curriculum. Such skills address the challenges of reproducibility, transparency, and increasingly large datasets. These skills benefit students both inside and outside academia. We acknowledge the challenges of incorporating these new skills, such as lack of time and the need to upskill staff, and we provide advice and resources to help tackle these. Moving toward a curriculum that puts reproducible data analysis at its core can be challenging, but given its many benefits, it is very much a journey worth undertaking.

References

Auker, L. A., & Barthelmess, E. L. (2020). Teaching R in the undergraduate ecology classroom: Approaches, lessons learned, and recommendations. *Ecosphere*, *11*(4), e03060. https://doi.org/10.1002/ecs2.3060

Azevedo, F., Parsons, S., Micheli, L., Strand, J. F., Rinke, E. M., Guay, S., Elsherif, M. M., Quinn, K. A., Wagge, J. R., Steltenpohl, C. N., Kalandadze, T., Vasilev, M. R., Oliveira, C. M., Aczel, B., Miranda, J. F., Baker, B. J., Galang, C. M. O., Pennington, C. R., Marques, T., … Forrt. (2019). *Introducing a Framework for Open and Reproducible Research Training (FORRT)* [Preprint]. Open Science Framework. https://doi.org/10.31219/osf.io/bnh7p

Baker, M. (2016). 1,500 scientists lift the lid on reproducibility. *Nature*, *533*(7604), 452–454. https://doi.org/10.1038/533452a

Baumer, B., Cetinkaya-Rundel, M., Bray, A., Loi, L., & Horton, N. J. (2014). R Markdown: Integrating A Reproducible Analysis Tool into Introductory Statistics. *Technology Innovations in Statistics Education*, *8*(1). https://doi.org/10.5070/T581020118

Bradford, L. (2018, October 11). Why All Employees Need Data Skills In 2019 (And Beyond). *Forbes*. https://www.forbes.com/sites/laurencebradford/2018/10/11/why-all-employees-need-data-skills-in-2019-and-beyond/?sh=7be3477d510f

Breznau, N., Rinke, E. M., Wuttke, A., Nguyen, H. H. V., Adem, M., Alvarez-Benjumea, A., Andersen, H. K., Auer, D., Azevedo, F., Balzer, D., Bauer, G., Bauer, P. C., Baumann, M., Baute, S., Bernauer, J., Berning, C., Berthold, A., Bethke, F. S., Biegert, T., … Van, J. (2022). Observing Many Researchers Using the Same Data and Hypothesis Reveals a Hidden Universe of Idiosyncratic Uncertainty. *Proceedings of the National Academy of Sciences*, *in press*, 23. https://doi.org/10.31222/osf.io/cd5j9

Broman, K. W., & Woo, K. H. (2018). Data Organization in Spreadsheets. *The American Statistician*, *72*(1), 2–10. https://doi.org/10.1080/00031305.2017.1375989

Button, K. S., Chambers, C. D., Lawrence, N., & Munafò, M. R. (2020). Grassroots Training for Reproducible Science: A Consortium-Based Approach to the Empirical Dissertation. *Psychology Learning & Teaching*, *19*(1), 77–90. https://doi.org/10.1177/1475725719857659

Chang, A. C., & Li, P. (2017). A Preanalysis Plan to Replicate Sixty Economics Research Papers That Worked Half of the Time. *American Economic Review*, *107*(5), 60–64. https://doi.org/10.1257/aer.p20171034

Committee on Envisioning the Data Science Discipline: The Undergraduate Perspective, Computer Science and Telecommunications Board, Board on Mathematical Sciences and Analytics, Committee on Applied and Theoretical Statistics, Division on Engineering and Physical Sciences, Board on Science Education, Division of Behavioral and Social Sciences and Education, & National Academies of Sciences, Engineering, and Medicine. (2018). *Data Science for Undergraduates: Opportunities and Options* (p. 25104). National Academies Press. https://doi.org/10.17226/25104

Dalrymple-Alford, E. C., & Budayr, B. (1966). Examination of Some Aspects of the Stroop Color-Word Test. *Perceptual and Motor Skills*, *23*(3_suppl), 1211–1214. https://doi.org/10.2466/pms.1966.23.3f.1211

Donoho, D. (2017). 50 Years of Data Science. *Journal of Computational and Graphical Statistics*, *26*(4), 745–766. https://doi.org/10.1080/10618600.2017.1384734

Eubank, N. (2016). Lessons from a Decade of Replications at the Quarterly Journal of Political Science. *PS: Political Science & Politics*, *49*(2), 273–276. https://doi.org/10.1017/S1049096516000196

Hardwicke, T. E., Mathur, M. B., MacDonald, K., Nilsonne, G., Banks, G. C., Kidwell, M. C., Hofelich Mohr, A., Clayton, E., Yoon, E. J., Henry Tessler, M., Lenne, R. L., Altman, S., Long, B., & Frank, M. C. (2018). Data availability, reusability, and analytic reproducibility: Evaluating the impact of a mandatory open data policy at the journal Cognition. *Royal Society Open Science*, *5*(8), 180448. https://doi.org/10.1098/rsos.180448

Knuth, D. E. (1984). Literate Programming. *The Computer Journal*, *27*(2), 97–111. https://doi.org/10.1093/comjnl/27.2.97

Moshontz, H., Campbell, L., Ebersole, C. R., IJzerman, H., Urry, H. L., Forscher, P. S., Grahe, J. E., McCarthy, R. J., Musser, E. D., Antfolk, J., Castille, C. M., Evans, T. R., Fiedler, S., Flake, J. K., Forero, D. A., Janssen, S. M. J., Keene, J. R., Protzko, J., Aczel, B., … Chartier,

C. R. (2018). The Psychological Science Accelerator: Advancing Psychology Through a Distributed Collaborative Network. *Advances in Methods and Practices in Psychological Science*, *1*(4), 501–515. https://doi.org/10.1177/2515245918797607

Munafò, M. R., Nosek, B. A., Bishop, D. V. M., Button, K. S., Chambers, C. D., Percie du Sert, N., Simonsohn, U., Wagenmakers, E.-J., Ware, J. J., & Ioannidis, J. P. A. (2017). A manifesto for reproducible science. *Nature Human Behaviour*, *1*(1), Article 1. https://doi.org/10.1038/s41562-016-0021

Nordmann, E., & DeBruine, L. (2022). *Applied Data Skills*. Zenodo. https://doi.org/10.5281/zenodo.6365078

Nosek, B. A., Spies, J. R., & Motyl, M. (2012). Scientific Utopia: II. Restructuring Incentives and Practices to Promote Truth Over Publishability. *Perspectives on Psychological Science*, *7*(6), 615–631. https://doi.org/10.1177/1745691612459058

Nuijten, M. B., Hartgerink, C. H. J., van Assen, M. A. L. M., Epskamp, S., & Wicherts, J. M. (2016). The prevalence of statistical reporting errors in psychology (1985–2013). *Behavior Research Methods*, *48*(4), 1205–1226. https://doi.org/10.3758/s13428-015-0664-2

Open Science Collaboration. (2015). Estimating the reproducibility of psychological science. *Science*, *349*(6251), aac4716. https://doi.org/10.1126/science.aac4716

PsyTeachR Team. (2022). *PsyTeachR*. https://psyteachr.github.io/

Quality Assurance Agency for Higher Education. (2019). *Subject Benchmark Statement: Psychology*. https://www.qaa.ac.uk/docs/qaa/subject-benchmark-statements/subject-benchmark-statement-psychology.pdf?sfvrsn=6935c881_15

Rouder, J. N. (2016). The what, why, and how of born-open data. *Behavior Research Methods*, *48*(3), 1062–1069. https://doi.org/10.3758/s13428-015-0630-z

Schacter, D. L. (1999). The seven sins of memory: Insights from psychology and cognitive neuroscience. *American Psychologist*, *54*, 182–203. https://doi.org/10.1037/0003-066X.54.3.182

Silberzahn, R., Uhlmann, E. L., Martin, D. P., Anselmi, P., Aust, F., Awtrey, E., Bahník, Š., Bai, F., Bannard, C., Bonnier, E., Carlsson, R., Cheung, F., Christensen, G., Clay, R., Craig, M. A., Dalla Rosa, A., Dam, L., Evans, M. H., Flores Cervantes, I., … Nosek, B. A. (2018). Many Analysts, One Data Set: Making Transparent How Variations in Analytic Choices Affect Results. *Advances in Methods and Practices in Psychological Science*, *1*(3), 337–356. https://doi.org/10.1177/2515245917747646

Simmons, J. P., Nelson, L. D., & Simonsohn, U. (2016). False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. In A. E. Kazdin (Ed.), *Methodological issues and strategies in clinical research (4th ed.).* (pp. 547–555). American Psychological Association. https://doi.org/10.1037/14805-033

Smith, S. A. (2022, June 28). Why we need data-literate citizens. *The Royal Society Blog*. https://royalsociety.org/blog/2022/06/envision-adrian-smith/

Stroop, J. R. (1935). Studies of Interference in serial visual reactions. *Journal of Experimental Psychology*, *18*, 643–662.

Taylor, A., Nelson, J., O'Donnell, S., Davies, E., & Hillary, J. (2022). *The Skills Imperative 2035: What does the literature tell us about essential skills most needed for work?* Slough: National Foundation for Educational Research.

Toelch, U., & Ostwald, D. (2018). Digital open science—Teaching digital tools for reproducible and transparent research. *PLOS Biology*, *16*(7), e2006022. https://doi.org/10.1371/journal.pbio.2006022

van Gog, T., Rummel, N., & Renkl, A. (2019). Learning how to solve problems by studying examples. In *The Cambridge handbook of cognition and education* (pp. 183–208). Cambridge University Press. https://doi.org/10.1017/9781108235631

Wagenmakers, E.-J., Wetzels, R., Borsboom, D., van der Maas, H. L. J., & Kievit, R. A. (2012). An Agenda for Purely Confirmatory Research. *Perspectives on Psychological Science*, *7*(6), 632–638. https://doi.org/10.1177/1745691612463078

Wicherts, J. M., & Bakker, M. (2012). Publish (your data) or (let the data) perish! Why not publish your data too? *Intelligence*, *40*(2), 73–76. https://doi.org/10.1016/j.intell.2012.01.004

Wickham, H. (2014). Tidy Data. *Journal of Statistical Software*, *59*, 1–23. https://doi.org/10.18637/jss.v059.i10

Zečević, K., Houghton, C., Noone, C., Lee, H., Matvienko-Sikar, K., & Toomey, E. (2021). *Exploring factors that influence the practice of Open Science by early career health researchers: A mixed methods study* (3:56). HRB Open Research. https://doi.org/10.12688/hrbopenres.13119.2