Developing Students' Written Communication Skills with Jupyter Notebooks

Conference or Workshop Item

How to cite:

Willis, Alistair; Charlton, Patricia and Hirst, Tony (2020). Developing Students' Written Communication Skills with Jupyter Notebooks. In: The 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20).

Version: Accepted Manuscript

Link(s) to article on publisher's website:
http://dx.doi.org/doi:10.1145/3328778.3366927

Developing Students' Written Communication Skills with Jupyter Notebooks

Alistair Willis
The Open University
Milton Keynes, U.K.
alistair.willis@open.ac.uk

Patricia Charlton
The Open University
Milton Keynes, U.K.
patricia.charlton@open.ac.uk

Tony Hirst
The Open University
Milton Keynes, U.K.
tony.hirst@open.ac.uk

ABSTRACT

Written communication skills are considered to be highly desirable in computing graduates. However, many computing students do not have a background in which these skills have been developed, and the skills are often not well addressed within a computing curriculum. For some multidisciplinary areas, such as data science, the range of potential stakeholders makes the need for communications skills all the greater. As interest in data science increases and the technical skills of the area are in ever higher demand, understanding effective teaching and learning of these interdisciplinary aspects is receiving significant attention by academics, industry and government in an effort to address the digital skills gap.

In this paper, we report on the experience of adapting a final year data science module in an undergraduate computing curriculum to help develop the skills needed for writing extended reports. From its inception, the module has used Jupyter notebooks to develop the students' skills in the coding aspects of the module. However, over several presentations, we have investigated how the cell-based structure of the notebooks can be exploited to improve the students' understanding of how to structure a report on a data investigation. We have increasingly designed the assessment for the module to take advantage of the learning affordances of Jupyter notebooks to support both raw data analysis and effective report writing.

We reflect on the lessons learned from these changes to the assessment model, and the students' responses to the changes.

CCS CONCEPTS

• **Social and professional topics** → *Computer science education*;
• **Applied computing** → *Interactive learning environments.*

KEYWORDS

Written communication, assessment, data science, Jupyter

ACM Reference Format:
Alistair Willis, Patricia Charlton, and Tony Hirst. 2020. Developing Students' Written Communication Skills with Jupyter Notebooks. In *The 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20), March 11–14, 2020, Portland, OR, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3328778.3366927

SIGCSE '20, March 11–14, 2020, Portland, OR, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-6793-6/20/03…$15.00
https://doi.org/10.1145/3328778.3366927

1 INTRODUCTION

It is becoming increasingly recognised that written communication skills are an important requirement for graduate students of all disciplines. However, many computing graduates lack these skills, and in fact do not always see them as an important part of their studies [4]. There have been several reports on efforts to increase the amount of writing in the computer science curriculum. These reports have highlighted the importance of integrating writing across the curriculum, rather than attempting to address these skills in separate, individual modules [9, 10, 14].

Data science is an emerging interdisciplinary discipline, which is now studied in many undergraduate and postgraduate Computer Science programmes. It is one of the areas seen as a core skill in high demand in the UK and elsewhere. Data science, as a domain of study, often sits between maths and computing and requires other academic and professional skills, such as report writing.

Recently, several reports have outlined the necessary competencies for data scientists, most of which identify communication skills as a key aspect of data science. For example [7]:

> A thoughtful data science program integrates communication-based opportunities and learning development throughout the whole of the curriculum rather than partitioning them into separate classes. Students should gain experience using oral, written, and visual modes to communicate effectively to a variety of audiences.

Also, the initial draft proposals from the ACM Taskforce on Data Science Education [6] list the desired communication competencies:

- Competencies
  - Evaluate an aspect of the technical literature relevant to data science
  - Produce a technical document for colleagues to use to guide technical development
  - Design and present a case to senior managers outlining a major initiative stemming from a data science investigation

Similarly, within the UK these skills have been highlighted by The Royal Society [26] and the Institute of Coding[1]. This latter is a recent government initiative bringing together a range of universities, industry, training providers and professional bodies to address the UK skills gap, one area of which concentrates on Data Science.

The study presented in this paper provides an insight into the affordances of Jupyter notebooks as a learning tool for data science, and our experience in using notebooks to develop students' skills in report writing. Our module's final assessment is an extended report on a data investigation of the students' choosing. As the

[1] www.instituteofcoding.org

module has developed, we have increasingly structured the assessment materials in a way that has supported the students' skills in constructing reports which discuss their data investigations.

This paper discusses our use of Jupyter notebooks in the module, how we have structured the assessment to develop the students' competence with report writing as well as assessing the technical learning outcomes, and how the students have responded to it.

2 JUPYTER IN DATA SCIENCE FOR PRACTICE AND TEACHING

Data science has a strong technical element, requiring practitioners to engage with a range of raw data and analysis of data in context. The many aspects of the data analysis pipeline (understanding what data to analyse and why, collecting and cleaning the data, analysing the data and, finally, reporting on the findings), require the practitioner to shift context at multiple points. This learning process is complex when working across a number of domains. It is easy for students to become lost in details and struggle with understanding the experimental steps and the impact of each step.

Jupyter notebooks are becoming a widely-used environment for developing and reporting on data science investigations, which help manage some of these difficulties. The notebooks are instantiated as a web service, in which a web page is divided into a number of individual *cells*. Each cell may contain formatted text, styled using a form of Markdown, or python code. In the latter case, the code is executed on the hosting server, and the output shown in the document (figure 1). The notebooks have the benefit of encouraging a strong narrative about the data investigation [18], and as result, promote reproducible research [13, 20, 24].

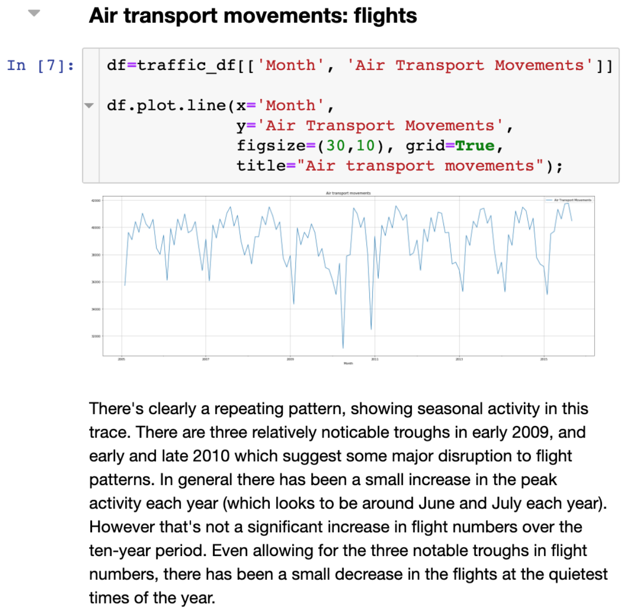
In the context of teaching, the notebook model allows code to be presented to the student, along with teaching commentary. For example, figure 1 shows cells taken from our module's teaching notes. In this snippet, python code is used to generate a graph from a dataset, and the subsequent text paragraph draws the students' attention to the key elements of the plot. In this case, the focus is on interpreting a plot. However, elsewhere in the module, the discussion might focus on explaining the code itself, referencing external resources and so on. Having seen the version of the code presented by the educators, students are then free to modify the code if they wish, and re-execute it *in situ.*

Within the teaching materials, we have also found many of the Jupyter extensions to provide valuable teaching opportunities. For example, Jupyter's code folding features allow self-assessment exercises for students to be neatly incorporated into the expository text. Figure 2 illustrates a self-assessment exercise for the students, in which our own solution is hidden, but can be revealed (and, of course, executed) within the notebook.

Although originally and primarily developed for data science, notebooks are widely used across Computing, and their value for teaching is becoming recognised across the curriculum [2, 11, 16].

3 THE TM351 DATA SCIENCE MODULE

The Computing and Communication department at the Open University in the UK runs a data science module aimed at third year undergraduates as part of the Computing and IT BSc programme. All authors of this paper are members of the module team. The

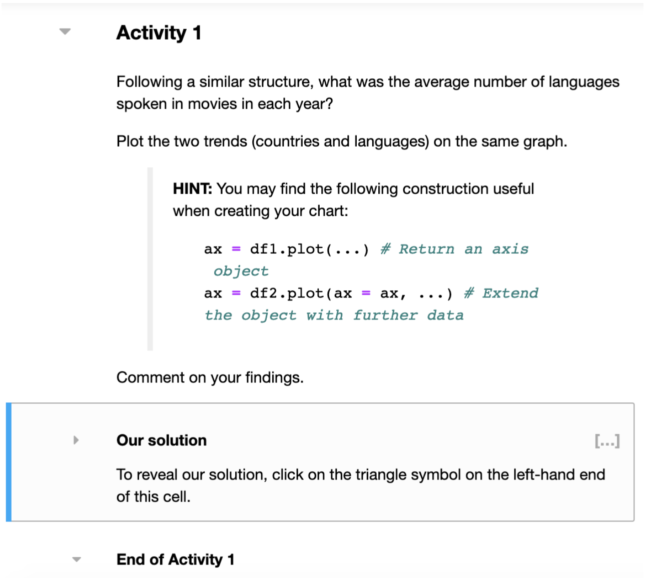


Air transport movements: flights

In [7]:
```
df=traffic_df[['Month', 'Air Transport Movements']]

df.plot.line(x='Month',
             y='Air Transport Movements',
             figsize=(30,10), grid=True,
             title="Air transport movements");
```

There's clearly a repeating pattern, showing seasonal activity in this trace. There are three relatively noticable troughs in early 2009, and early and late 2010 which suggest some major disruption to flight patterns. In general there has been a small increase in the peak activity each year (which looks to be around June and July each year). However that's not a significant increase in flight numbers over the ten-year period. Even allowing for the three notable troughs in flight numbers, there has been a small decrease in the flights at the quietest times of the year.

Figure 1: Executed code and explanatory cells in Jupyter



Activity 1

Following a similar structure, what was the average number of languages spoken in movies in each year?

Plot the two trends (countries and languages) on the same graph.

> **HINT:** You may find the following construction useful when creating your chart:
>
> ```
> ax = df1.plot(...) # Return an axis
>  object
> ax = df2.plot(ax = ax, ...) # Extend
> the object with further data
> ```

Comment on your findings.

Our solution [...]

To reveal our solution, click on the triangle symbol on the left-hand end of this cell.

End of Activity 1

Figure 2: Self-assessment exercises within Jupyter

Open University is a distance learning institution, and so the module is delivered remotely to students who are typically studying part time while in employment. The module lasts for 9 months, and is equivalent to a quarter of a full time student's final year study. The first presentation started in February 2016, and the most recent started in October 2018.

The module is delivered via two main media: an online textbook which focusses mainly on theoretical aspects of the work, and a collection of notebooks, which focus on the practical aspects.

The notebook server is currently delivered to students as a Virtual Machine [12], allowing a consistent experience regardless of each student's particular platform.

The coding language of instruction is Python (the module itself is delivered in English), but the structure of the notebooks means that the main teaching points are language agnostic: the lessons in this paper could apply equally to the use of R, Julia or any language that can be run incrementally and for which there exists a notebook kernel. It is the experimental context which is important for data science, and which supports the development of a narrative that can be used to work on the students' report writing.

The module consists of two main pieces of continuous assessment which are manually marked by tutors. These assessments are intended to be primarily formative, with a strong focus on tutor feedback. Both are submitted as Jupyter notebooks. These assessments will be discussed in more detail in section 4. An important element of the module is that all datasets used are publicly available, open data sets (from sites such as the WHO data repository[2] or government datasets[3]). This is to demonstrate to students the difficulties found in real-world data investigations, and to emphasise that the techniques they taught are appropriate for such data.

The main piece of assessment for the module is a report detailing the students' attempt at an exploratory data analysis [27], which should cover their approaches to cleaning, storing, analysing and visualising the datasets. Students are provided with (at least) two datasets, and are required to identify two questions to ask of that data. The report should be targetted at an appropriate general audience for those questions. For example, in one presentation, students were given a dataset detailing the financial and legal statuses and attainment of schools across the UK, and the 2011 UK census data. Students typically answered questions exploring the relationship between school performance and income levels in different areas of the UK, or between school provision and ethnic profiles, and so on.

The final report is submitted as a single Microsoft Word document, reflecting the norm in industry, and is strictly capped at 3,000 words. Technical work is submitted as a Jupyter notebook in an appendix which does not contribute to the final mark, and is included for validation only.

The final report is marked according to five criteria:

(1) Understanding of legal issues,
(2) Identifying practical questions from the data, and answering them,
(3) Preparing and storing the data in an appropriate manner,
(4) Using appropriate techniques to analyse the data, and
(5) The quality of the final report.

In each case, the students are assessed both on their ability to carry out that aspect of the work, as well as their ability to make a critical assessment of their decisions. For example, the ability to clean a dataset (such as by writing appropriate python to fill in missing values) is assessed, but so is the student's ability to explain (unprompted) what impact his or her choices of cleaning techniques are likely to have had on the final analysis.

From the first presentation of TM351, we have had a policy that the mark scheme should be available to students, and so all students have the criteria and their interpretation available to them. Despite this, in the early presentations of the module, students still appeared to be struggling with developing a final report.

4 PREPARING STUDENTS FOR REPORT WRITING

Over the presentations of the module, it became clear to the module team that the students on the module had not had the level of support in formal writing in the curriculum that they needed to write a well-focussed report on their data investigations. We therefore needed to identify ways of developing this skill within the module, while ensuring that the academic level of the module remained appropriate for final year undergraduates.

Students are required to complete two pieces of assessment over the course of the module. These are intended primarily as formative assessment, allowing the students to receive feedback on their work. To start developing the writing skills as early as possible in the module, we divided the first piece of continuous assessment (submitted around one third of the way through the module) into two closely related parts. The first part is a heavily guided data investigation in which each of the steps is expressed as a programming exercise. The second then provides a notebook with a skeleton investigation structure for the students to carry out a further investigation using similar techniques on different datasets.

The skeleton format presented in the second part of the question should form a generic structure for the students' investigations which can then be used as the basis for the final report. By introducing this structure early in the module, tutors can easily identify where students are having difficulty either with some of the key technical tasks (such as how to clean and reshape the data), or how to justify the technical choices they have made.

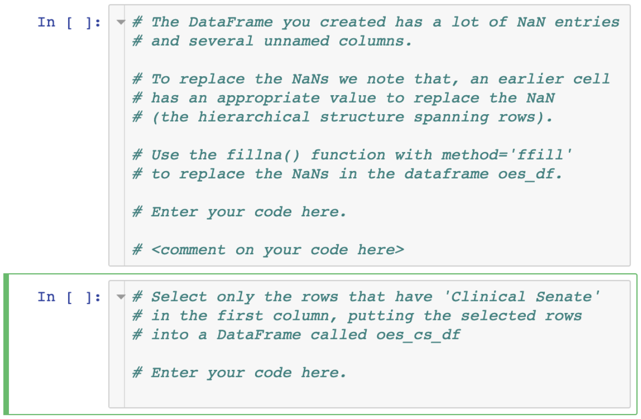
4.1 Part 1: Focussing on the data

As an experience report, in this paper we are concerned with demonstrating how our pedagogical approach has *changed* over the lifetime of the module. The module has now run for four presentations, and our design of the assessment has changed in response to feedback from students (discussed further in section 6).

Over the four presentations, we have moved towards a set of notebooks which more and more closely reflect the structure of the final report that the students are expected to produce at the end of the module (section 3). In the first presentation, the first piece of student assessment was a traditional set of python exercises, developed for the purpose of ensuring that the students had the basic programming skills required to manipulate the datasets. For example, figure 3 shows a collection of cells from the module's first presentation. This figure shows the use of highly directed tasks: the focus was on trying to get the student to show they can translate a task into python, rather than to reflect on the task itself.

During the first two presentations of the module, it became clear that this style of assessment was not adequately preparing the students for the final report. We have therefore moved towards questions which focus much more clearly on the broader *purpose* of each programming exercise, rather than the code itself. Figure 4 shows comparable cells from the fourth presentation: rather than focussing on the python code, the questions focus on the *data*,

[2] https://www.who.int/gho/database/
[3] https://www.data.gov/

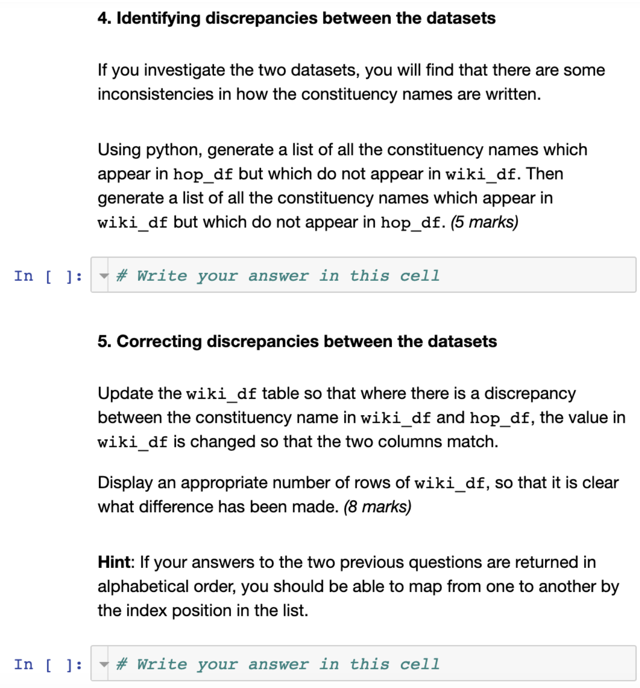


```
In [ ]:  # The DataFrame you created has a lot of NaN entries
         # and several unnamed columns.

         # To replace the NaNs we note that, an earlier cell
         # has an appropriate value to replace the NaN
         # (the hierarchical structure spanning rows).

         # Use the fillna() function with method='ffill'
         # to replace the NaNs in the dataframe oes_df.

         # Enter your code here.

         # <comment on your code here>
```

```
In [ ]:  # Select only the rows that have 'Clinical Senate'
         # in the first column, putting the selected rows
         # into a DataFrame called oes_cs_df

         # Enter your code here.
```

Figure 3: Assessment cells from the first presentation

with the python exercise being framed in terms of the task that the python is intended to *achieve.*



4. Identifying discrepancies between the datasets

If you investigate the two datasets, you will find that there are some inconsistencies in how the constituency names are written.

Using python, generate a list of all the constituency names which appear in `hop_df` but which do not appear in `wiki_df`. Then generate a list of all the constituency names which appear in `wiki_df` but which do not appear in `hop_df`. *(5 marks)*

```
In [ ]:  # Write your answer in this cell
```

5. Correcting discrepancies between the datasets

Update the `wiki_df` table so that where there is a discrepancy between the constituency name in `wiki_df` and `hop_df`, the value in `wiki_df` is changed so that the two columns match.

Display an appropriate number of rows of `wiki_df`, so that it is clear what difference has been made. *(8 marks)*

Hint: If your answers to the two previous questions are returned in alphabetical order, you should be able to map from one to another by the index position in the list.

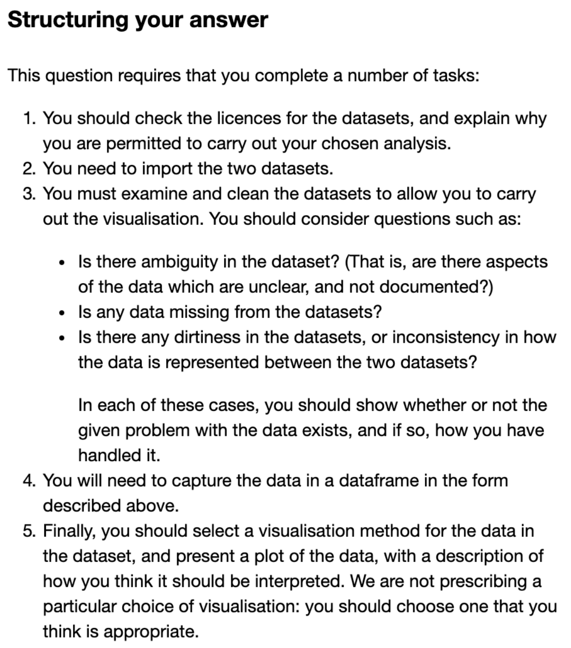
```
In [ ]:  # Write your answer in this cell
```

Figure 4: Assessment cells from the fourth presentation

In the first presentation, the task is phrased in terms of replacing null values in a pandas dataframe. In the fourth, the task is phrased in terms of inconsistencies in UK parliamentary constituencies. What is needed from the student is the same in both cases, but the focus has been explicitly shifted from the code to the data.

4.2 Part 2: Focussing on the report

Having been presented with a set of techniques in part 1 of the assessment, in the second part, students are encouraged to follow a similar pattern on a new dataset. The students are given the standard preamble shown in figure 5.



Structuring your answer

This question requires that you complete a number of tasks:

1. You should check the licences for the datasets, and explain why you are permitted to carry out your chosen analysis.
2. You need to import the two datasets.
3. You must examine and clean the datasets to allow you to carry out the visualisation. You should consider questions such as:
    - Is there ambiguity in the dataset? (That is, are there aspects of the data which are unclear, and not documented?)
    - Is any data missing from the datasets?
    - Is there any dirtiness in the datasets, or inconsistency in how the data is represented between the two datasets?

    In each of these cases, you should show whether or not the given problem with the data exists, and if so, how you have handled it.
4. You will need to capture the data in a dataframe in the form described above.
5. Finally, you should select a visualisation method for the data in the dataset, and present a plot of the data, with a description of how you think it should be interpreted. We are not prescribing a particular choice of visualisation: you should choose one that you think is appropriate.

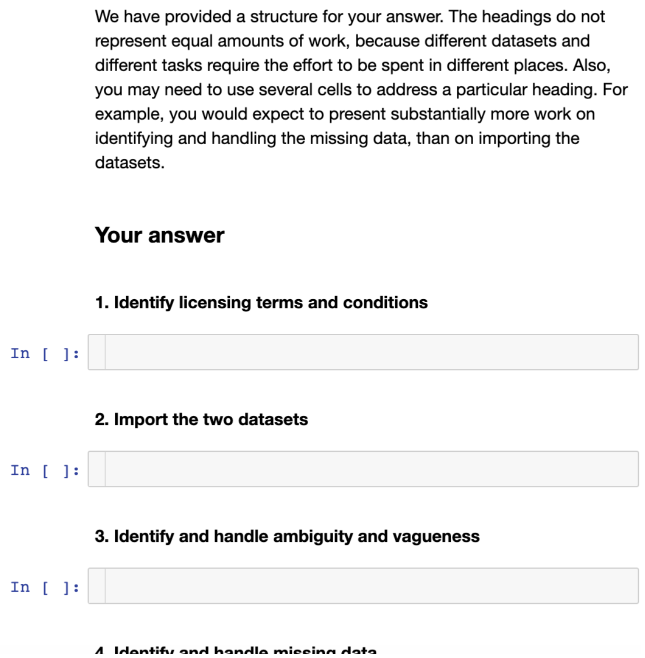
Figure 5: Preamble for second part of the question

Importantly, the preamble emphasises the need for explanation, with phrases such as “explain why” (task 1), “show how you have handled it” (point 3), and “provide a description [of your visualisation]” (point 5). These instructions aim to repeatedly emphasise the importance of the students’ judgement in their responses, as well as giving credit for appropriate explanations of their work. We also use these to show how the coding aspects should be integrated into, and used to support, the discursive aspects.

Figure 6 illustrates the first few cells of the notebook actually given to students for the second part of the assessment. Each of the headings corresponds to a typical task and paragraph in a data investigation of the form being developed. As figure 6 shows, students are not given any indication of the specific tasks that need to be carried out. Rather, a list of general tasks is given, and the students are expected to apply their experience from the first part of the question to the second. The complete list is:

(1) Identify licensing terms and conditions
(2) Import the two datasets
(3) Identify and handle ambiguity and vagueness
(4) Identify and handle missing data
(5) Identify and handle inconsistent or dirty data
(6) Put the data into an appropriate form for plotting
(7) Visualise the data
(8) Interpret your plot

Finally, an important aspect of this assessment is indicated in the guidance to markers. Markers are explicitly told that the more free-form nature of the second part of the assessment suggests



We have provided a structure for your answer. The headings do not represent equal amounts of work, because different datasets and different tasks require the effort to be spent in different places. Also, you may need to use several cells to address a particular heading. For example, you would expect to present substantially more work on identifying and handling the missing data, than on importing the datasets.

Your answer

1. Identify licensing terms and conditions

```
In [ ]:
```

2. Import the two datasets

```
In [ ]:
```

3. Identify and handle ambiguity and vagueness

```
In [ ]:
```

4. Identify and handle missing data

Figure 6: Jupyter cells indicating structure for an exploratory data investigation (second part of the assessment)

that there is no "model solution" or similar to be used as reference. Rather, markers are told:

> Remember that this question is not attempting to find some sort of "correct" answer from the students. Rather, the question is aimed at getting the students to recognise the vagaries and problems of tasks that occur in genuine data investigations. Credit should be awarded for recognising the issues and proposing appropriate or realistic solutions, rather than steering students towards a single gold standard. In particular, we will meet many techniques for handling these uncertainties through the module: at this point it is more important that the students recognise the potential problems and have the confidence to suggest solutions, even if those solutions are not what we would hope for in the [final report].

The open-ended nature here is intended to support students' communication and technical prowess in tandem, rather than treating them as different aspects of the work.

5 COMMENTS ON THE PEDAGOGICAL APPROACH

While Jupyter notebooks are not the only tool used when teaching and learning data science, part of the design approach by the team was to use notebooks as a means to support students with the incremental investigation of data, using the notebooks approach as scientific logbook. At the time of the first presentation of the module in 2016, this approach was relatively new. Since then, there has been a rapid increase in the use of notebooks, but relatively little analysis of their potential from a pedagogical perspective.

Table 1: Analysis of learning design approach

Learning Activity	Description
Interdisciplinary context and purpose setting with real data problems	Introducing Data Science
Authenticity: active learning	Case-based approach throughout - working with real data
Guided interactive exploration	Cell by Cell exploration of an example data challenge in first part of assessment
Active learning	Second part of assessment follows the structure of the first, but more open ended
Active independent learning	Final assessment: Unsupported case study, following previous structures
Reflections and feedback	Required as an element of the final report

One of the great advantages afforded by the notebooks is that the platform both provides a valuable learning environment, as well as being the chosen tool for many practitioners in the data science field. Education research across the sector has shown the value of such authentic and interactive learning experiences. They confirm the research findings of Bransford et al. [1], Seligman et al. [23] and Scardamalia and Bereiter [22], who found that when *active* pedagogical approaches are used, the authentic learning settings potentially result in the development of interdisciplinary problem solving skills and resilience. Both interdisciplinary problem solving and resilience are central when learning to become a data analyst.

Dewey [8], Piaget [19], Papert [17] and others have written about the power of learning through the experiential process of creating tangible objects. A key observation by Sill [25] about engagement with critical thinking is its complexity and thus may be resisted by learners. However, when reexamining critical thinking through Chandrasekharan's [3] "Tinker Media" environments, some of the barriers are alleviated. The process of simulation in a Tinker Media setting provides the conditions of experimentation and timely feedback. It is exactly these conditions that using Jupyter notebooks environment have the potential to offer students learning to be data analysts - bringing both raw data analysis and narrative together through the notebooks' close proximity of data and narrative.

Following Rusk et al. [21] and Laurillard et al. [15], table 1 shows a proposal for learning design using Jupyter notebooks. The scaffolded approach to developing the students' report writing skills is reflected in the table.

6 STUDENT RESPONSES

It is difficult to evaluate the success or otherwise of an intervention such as the one described in this paper. However, responses to our institution's standard module satisfaction survey indicates a considerable improvement in the student satisfaction on their assessment. The response rate was 30.2% for the first presentation and 21.8% for the fourth presentation (following the redesigned assessment) out of cohorts of 232 and 303 students respectively. Responses are obtained only from students who complete the module.

Students are asked for comments in free text, as well as for a number of Likert-style questions. Following the first presentation of the module, several students raised concerns about how they felt unprepared for the report in the final assessment:

> "Incredibly simple examples with very clean data is good way to start learning. However the [final report] then throws you in the deep end."
>
> "this is by far my worst module solely because of the [final report]. Guidance was poor"
>
> "perhaps more practice (far more) on... reporting (I feel this has been underused in the course- making us doing the [final report] much harder."

One comment also supports Cilliers' [5] suggestion that students do not always appreciate the value of these tasks:

> "a huge proportion of the marks for the [final assessment] seemed to be for elements which were not directly relevant to the course content (eg. how well you can write a report)."

In the student comments for the later presentations, there appeared to be much more appreciation for the way the notebooks had been structured to develop the students' particular skills. The following comments were taken from the feedback on the third presentation of the module, after the more structured approach to the assessment had been implemented:

> "The [notebooks] that guided the user through completing tasks was very helpful to completing the tasks required for the module."
>
> "particularly valued the notebooks and walk-through examples"

Both of these quotes emphasise the benefits achieved from the greater level of guidance in the notebooks, and the relevance of that guidance towards the later assessment. In addition, an benefit of the notebooks is that students are able to take more *ownership* of their work. Rather than the notebooks being a static representation of the teaching materials, students are able to adapt them for their own understanding. The comment:

> "I personally found the notebooks to be the most helpful to my learning on this module, they not only helped explain concepts but showed examples before you could try it yourself"

recognises that the notebook itself can be adapted, rather than simply being a description of work to be investigated elsewhere.

Finally, one question in the post-module survey asks students for their satisfaction with the assessment. The student responses for the first (February 2016) and fourth (October 2018) presentations are shown in figure 7. These appear to show a substantial increase in the students' satisfaction with our assessment strategy across the reporting period. Note that the student survey question has been changed slightly between the two presentations as a result of institutional decisions. For the 2016 presentation, students were asked whether they agreed with the statement "Overall, I was satisfied with the assessment on this module, " while for the 2018 presentation, students were asked whether they agreed with the statement "It was obvious how the module materials related to the assessed tasks on this module." However, these two questions are the closest equivalents in the two surveys.

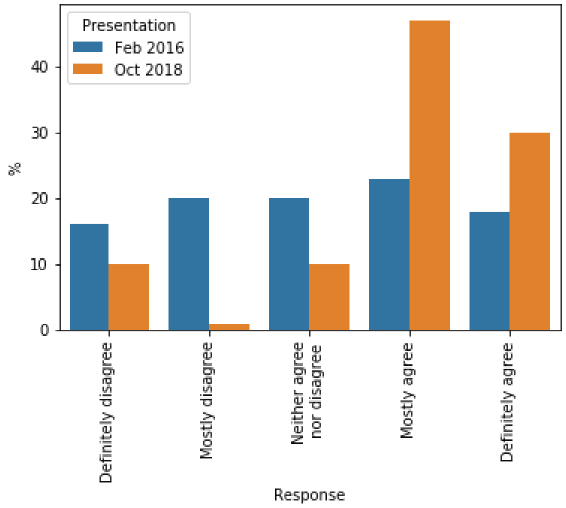


Figure 7: Students' reported agreement on whether they considered the module assessment to be satisfactory

7 CONCLUSIONS

Our aim with the assessment on this module has been to provide students with the opportunity to improve their written communication skills for data science. We believe that the Jupyter environment has supported this aim, and the students' improved view of the module's assessment strategy appears to support this. The strategy that we have used does, of course, encourage a fairly uniform structure of report amongst the students. However, we feel that for the purpose of developing the skills of creating reports, this level of scaffolding is appropriate. The aim in this module is to develop the students' ability to write up a data investigation, not to be to write in several different styles. In future work, we will investigate whether there has been a specific improvement in the marks awarded for the writing criterion of the final assessment.

A further benefit of the notebooks that has become clear, is that as students take ownership of their notebooks and the level of structure provided by the module team is reduced, students are able to express themselves more individually. As notebooks become more widely used throughout computing, this highlights the opportunity to embed writing techniques across the curriculum.

ACKNOWLEDGMENTS

This work has been completed in the context of The Institute of Coding, an initiative funded by the UK Office for Students. The authors would also like to thank the anonymous reviewers for their helpful comments and suggestions.

REFERENCES

[1] John D Bransford, Ann L Brown, and Rodney R Cocking. 2000. *How people learn.* Vol. 11. Washington, DC: National academy press.

[2] Robert J. Brunner and Edward J. Kim. 2016. Teaching data science. *Procedia Computer Science* 80 (2016), 1947–1956.

[3] Sanjay Chandrasekharan. 2009. Building to discover: a common coding model. *Cognitive Science* 33, 6 (2009), 1059–1086.

[4] Charmain B Cilliers. 2012. Student perception of academic writing skills activities in a traditional programming course. *Computers & Education* 58, 4 (2012), 1028–1041.

[5] Charmain B. Cilliers. 2012. Student perception of academic writing skills activities in a traditional programming course. *Computers & Education* 58, 4 (2012), 1028–1041.

[6] Andrea Danyluk, Paul Leidig, Lillian Cassel, and Christian Servin. 2019. ACM Task Force on Data Science Education: Draft Report and Opportunity for Feedback. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19).* ACM, New York, NY, USA, 496–497. https://doi.org/10.1145/3287324.3287522

[7] Richard D De Veaux, Mahesh Agarwal, Maia Averett, Benjamin S Baumer, Andrew Bray, Thomas C Bressoud, Lance Bryant, Lei Z Cheng, Amanda Francis, Robert Gould, et al. 2017. Curriculum guidelines for undergraduate programs in data science. *Annual Review of Statistics and Its Application* 4 (2017), 15–30.

[8] John Dewey. 1916. Education and democracy.

[9] Katrina Falkner and Nickolas JG Falkner. 2012. Integrating communication skills into the computer science curriculum. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education.* ACM, 379–384.

[10] Harriet J Fell, Viera K Proulx, and John Casey. 1996. Writing across the computer science curriculum. In *ACM SIGCSE Bulletin*, Vol. 28. ACM, 204–209.

[11] Ben Glick and Jens Mache. 2018. Using jupyter notebooks to learn high-performance computing. *Journal of Computing Sciences in Colleges* 34, 1 (2018), 180–188.

[12] Chris Holdgraf, Aaron Culich, Ariel Rokem, Fatma Deniz, Maryana Alegro, and Dani Ushizima. 2017. Portable learning environments for hands-on computational instruction: Using container-and cloud-based technology to teach data science. In *Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact.* ACM, 32.

[13] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian E. Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica B. Hamrick, Jason Grout, and Sylvain Corlay. 2016. Jupyter Notebooks-a publishing format for reproducible computational workflows.. In *ELPUB*. 87–90.

[14] Clifton Kussmaul. 2005. Using agile development methods to improve student writing. *Journal of Computing Sciences in Colleges* 20, 3 (2005), 148–156.

[15] Diana Laurillard, Patricia Charlton, Brock Craft, Dionisios Dimakopoulos, Dejan Ljubojevic, George Magoulas, Elizabeth Masterman, Roser Pujadas, Edgar A. Whitley, and Kim Whittlestone. 2013. A constructionist learning environment for teachers to model learning designs. *Journal of Computer Assisted Learning* 29, 1 (2013), 15–30.

[16] Keith O'Hara, Douglas Blank, and James Marshall. 2015. Computational notebooks for AI education. In *The Twenty-Eighth International Flairs Conference.*

[17] Seymour Papert. 1990. Children, computers and powerful ideas.

[18] Fernando Perez and Brian E. Granger. 2015. Project Jupyter: Computational narratives as the engine of collaborative data science. *Retrieved September* 11, 207 (2015), 108.

[19] Jean Piaget. 2013. *Play, dreams and imitation in childhood.* Routledge.

[20] Bernadette M Randles, Irene V Pasquetto, Milena S Golshan, and Christine L Borgman. 2017. Using the Jupyter notebook as a tool for open science: An empirical study. In *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL).* IEEE, 1–2.

[21] Natalie Rusk, Mitchel Resnick, Robbie Berg, and Margaret Pezalla-Granlund. 2008. New pathways into robotics: Strategies for broadening participation. *Journal of Science Education and Technology* 17, 1 (2008), 59–69.

[22] Marlene Scardamalia and Carl Bereiter. 2006. *Knowledge building: Theory, pedagogy, and technology.* na.

[23] Martin EP Seligman, Randal M Ernst, Jane Gillham, Karen Reivich, and Mark Linkins. 2009. Positive education: Positive psychology and classroom interventions. *Oxford review of education* 35, 3 (2009), 293–311.

[24] Helen Shen. 2014. Interactive notebooks: Sharing the code. *Nature News* 515, 7525 (2014), 151.

[25] David J Sill. 1996. Integrative thinking, synthesis, and creativity in interdisciplinary studies. *The Journal of General Education* 45, 2 (1996), 129–151.

[26] The Royal Society. 2019. Dynamics of data science skills: How can all sectors benefit from data science talent? https://royalsociety.org/-/media/policy/projects/dynamics-of-data-science/dynamics-of-data-science-skills-report.pdf

[27] Chong Ho Yu. 1977. Exploratory data analysis. *Methods* 2 (1977), 131–160.