

Processing Student Responses to STACK Questions in Moodle Quizzes

Jonas Lache^{1,2}

Abstract: When students complete STACK questions in Moodle, a lot of information is stored and can be accessed via the Moodle quiz report. It is sensible to use this information to generate visualisations and descriptive statistics to gain insight into how students use STACK questions in Moodle quizzes. This paper focuses on ideas for processing the export files and extracting information from the data using Python pandas and regular expressions. After processing, the results can be analysed. This paper is aimed at researchers and practitioners looking for effective ways to analyse Moodle quiz reports.

Keywords: STACK; Moodle; Data; Procession; Analysis; Python

1 Introduction

The data available in Moodle can be used for a variety of purposes. One possibility is to use it to improve STACK questions, for example by gaining information about the actual answers given by students and thus adding more nodes to the response trees to catch common mistakes. For this purpose, the “Basic question use report” built into the STACK plugin is a useful choice (see <https://docs.stack-assessment.org/en/Authoring/Reporting/>). In addition, Moodle data can be used to gain insight into students’ question usage patterns.

There are several ways to get data from Moodle. As well as the more general Moodle logs, which include information such as how often Moodle pages and activities have been visited, see [ALK19], there is also data specific to Moodle quizzes. In the quiz options the “Grades”, “Statistics” and “Responses” reports are available for download. While all of these contain information about students’ quiz attempts, this paper will concentrate on how to process the “Responses” output files in order to use them for data analysis. First, these files and how to obtain them is explained. Then the processing of the “Responses” files is discussed. The paper ends with an outlook on possible next steps.

¹ Ruhr-Universität Bochum, Fakultät für Mathematik, Postfach 70, 44780 Bochum, Germany; jonas.lache@ruhr-uni-bochum.de

² Hochschule Ruhr West – University of Applied Sciences, E-Learning – Dezernat III, Duisburger Str. 100, 45479 Mülheim an der Ruhr, Germany; jonas.lache@hs-ruhrwest.de

2 The “Responses” report file

A “Responses” report file is basically a table where each row contains information about a single step of work done by a student (see Fig. 1). If the adaptive mode is enabled in the Moodle quiz, students can answer the STACK question multiple times in one attempt by clicking on a “Check” button. In this case, each row of the table represents one click on the “Check” button. Otherwise, each row represents one attempt. In the following we will concentrate on the first case. To download the “Responses” file (e.g. in CSV format), teachers can click on the “Results” tab within a Moodle quiz activity. The “Responses” option should then appear. Before starting the download, it is possible to specify details about the export file, for example, which attempts should be included.

	A	B	C	D	E	F	
1	Alias	State	Started on	Completed	Time taken	Grade/10.00	Response 1
2	BO7JMWWMuX43Y8zi		5 April 2023 8:49 AM	5 April 2023 8:53 AM	4 mins 4 secs		Seed: 1646667833;
3	BO7JMWWMuX43Y8zi		5 April 2023 8:49 AM	5 April 2023 8:53 AM	4 mins 4 secs		Seed: 1646667833;
4	BO7JMWWMuX43Y8zi		5 April 2023 8:49 AM	5 April 2023 8:53 AM	4 mins 4 secs		Seed: 1646667833;
5	BO7JMWWMuX43Y8zi		5 April 2023 8:49 AM	5 April 2023 8:53 AM	4 mins 4 secs		Seed: 1646667833;
6	BO7JMWWMuX43Y8zi		5 April 2023 8:49 AM	5 April 2023 8:53 AM	4 mins 4 secs		Seed: 1646667833;
7	BO7JMWWMuX43Y8zi		5 April 2023 8:49 AM	5 April 2023 8:53 AM	4 mins 4 secs		Seed: 1646667833;
8	BO7JMWWMuX43Y8zi		5 April 2023 8:49 AM	5 April 2023 8:53 AM	4 mins 4 secs		Seed: 1646667833;
9	BO7JMWWMuX43Y8zi	Finished	5 April 2023 8:49 AM	5 April 2023 8:53 AM	4 mins 4 secs	10.00	Seed: 1646667833;
10	yLtYtNN3MJBNkpkz	Finished	5 April 2023 8:53 AM	5 April 2023 8:57 AM	4 mins 18 secs	0.00	Seed: 1579016987;
11							

Fig. 1: Example of a pseudonymised “Responses” file.

By default, the file contains personal information about the students, such as an email address. As I am working with pseudonymised data, the first column in Fig. 1 contains a unique alias for each student. Using these pseudonyms, the data can be matched with other data (e.g. the results of a survey), making it potentially useful for research purposes. The second column of the table contains the keyword “Finished” if the student finished the attempt in that row. In the third to fifth columns we have information about the time when the attempt was started and finished, and how much time it took. Note that the file does not contain time stamps for each click on the “Check” button, but only data relating to the start and completion of the quiz. The sixth column contains the grade achieved by the student. Since the grade is only given when an attempt is finished, only rows with value “Finished” in the “State” column can contain a grade. The last and most interesting column is the “Response 1” column. If there are multiple questions in a quiz, columns “Response 2”, “Response 3”, etc. are created. The column contains STACK specific information such as the actual expression within the input fields and which potential response trees were active. It also includes the seed for the question used and information about the nodes of each response tree that the students went through – the answer notes. An entry in this column might look like this:

```
Seed: 325231; ans1: x^2 [score]; ans2: pi [score]; prt1: # = 0 | prt1-1-F;
prt2: # = 1 | prt2-1-T; prt3: !
```

Although this string is not difficult to understand and well documented (see <https://docs.stack-assessment.org/en/Authoring/Reporting/>), it is not yet in a form that allows immediate analysis of the data. Therefore, some processing is required to bring the information from the column into a better form.

3 Procession

To process “Responses” files, Python can be used. A Jupyter Notebook (<https://jupyter.org/>) can be used to keep the workspace clean and tidy. For data analysis, data manipulation and scientific computing, the Python libraries `pandas` and `numpy` are required. In addition, the `re` library is required for the use of regular expressions.

After importing a “Responses” file, we can start extracting information from the “Response” column. But before we do this, it is useful to prepare the data. For example, the entries in the “Time taken” column are no numbers or in an ISO format for durations. Instead the column contains strings like “4 mins 18 secs”, which Python does not understand. To work around this problem, we can write a Python function that converts these strings to seconds and apply it to the “Time taken” column. Note the use of regular expressions (line 6). The dictionary in the first line, however, contains conversion factors for each unit of time (e.g. one hour equals 3600 seconds):

```

1 time_values = {"secs":1, "sec":1, "mins":60, "min":60, "hours":3600,
2               "hour":3600, "days":86400, "day":86400}
3
4 def string_time_to_sec(time_str):
5     total_seconds = 0
6     time_parts = re.findall(r"(\d+)\s?(\w+)", time_str)
7     for time_part in time_parts:
8         value, unit = time_part
9         total_seconds += int(value) * time_values[unit]
10    return total_seconds
11 # Apply function to "Time taken" column:
12 df["Seconds taken"] = df["Time taken"].apply(string_time_to_sec)

```

Once this procession is complete, we can move to the “Response” column and extract information from it. But what is the best way to store the extracted information? One approach that has worked for me is to add a new column to the table for each piece of information that is extracted. This has the advantage that we retain the previous structure of the table, no information is lost and the overall view of the data is clear. There are many examples of information that can be of interest and extracted from the “Response” column: The score achieved by students in a particular sub-task, information about whether a particular potential response tree was active, or the value of a particular STACK input field. If STACKrate has been used – an evaluation tool that allows students to rate STACK questions using a star rating principle, see [LM22] – the results of the evaluation can be

extracted using this approach as well. To extract information from the “Response” column, we can use the same method as in the previous example with the “Time taken” column: We first write a Python function and then apply it to the “Response” column.

Let us start with a simple example: To find out whether a particular potential response tree has been active, we can use the following function. It is very simple, since all we have to do is check whether the string in the “Response” column contains a particular substring:

```
1 def is_prt_active(response_str, prt_name):
2     return prt_name + " : # =" in response_str
```

The function must then be applied to the “Response 1” column:

```
1 df["prt2 active"] = df["Response 1"].apply(lambda x: is_prt_active(x, "prt2"))
```

The result is a new column named “prt2 active” containing the value “True” if the potential response tree “prt2” was active in the work step corresponding to the row, and “False” if it was not (see Fig. 2).

	G	H	
en	Grade/10.00	prt2 active	Response 1
244		False	Seed: 1646667833; ans1: {
244		False	Seed: 1646667833; ans1: {
244		False	Seed: 1646667833; ans1: {
244		True	Seed: 1646667833; ans1: {
244		True	Seed: 1646667833; ans1: {
44		True	Seed: 1646667833; ans1: {
44		True	Seed: 1646667833; ans1: {
44	10.00	True	Seed: 1646667833; ans1: {
58	0.00	False	Seed: 1579016987

Fig. 2: “Responses” file with new column “prt2 active”.

It is also possible to add columns like this simultaneously for each potential response tree using a for loop:

```
1 for prt in ["prt1", "prt2", "prt3"]
2     df[prt + " active"] = df["Response 1"].apply(lambda x: is_prt_active(x, prt))
```

Another example of a piece of information to be extracted from the “Response 1” column is the score that students have achieved in a particular sub-task. STACK questions can consist of several potential response trees, so we might be interested in the score achieved in, for example, the response tree named “prt1”. To find out the score achieved in “prt1” we can apply the following Python function to our “Response 1” column:

```

1 def get_prt_score(response_str, prt):
2     pattern = r"{:} #: ([0-9]+(?:\.[0-9]+)?)".format(prt)
3     match = re.search(pattern, response_str)
4     if match:
5         return float(match.group(1))
6     else:
7         return np.nan
8 # Apply function to "Response 1" column:
9 df["prt1 score"] = df["Response 1"].apply(lambda x: get_prt_score(x, "prt1"))

```

Again, we can create columns like this for more than one potential response tree at a time using a for loop:

```

1 for prt in ["prt1", "prt2", "prt3"]:
2     df[prt + " score"] = df["Response 1"].apply(lambda x: get_prt_score(x, prt))

```

4 Outlook

Using the method discussed above, many types of information can be extracted from the report files available in Moodle. The approach of adding a new column to the table for each piece of information extracted makes it possible to analyse the data. For example, it is now possible to calculate parameters such as averages and frequencies. However, before doing this, it may be useful to reduce the number of rows in the “Responses” file to one row for each attempt. The groupby operation from the pandas library can be used to do this:

```

1 df.groupby(["Alias", "Started on"]).agg({
2     "State": "last",
3     # Decide method for every column
4     ...
5     "Response 1": " <NEW ROW> ".join,
6 })

```

In the list ["Alias", "Started on"] we specify that Python should consider the rows to be part of the same attempt of the Moodle quiz that have the same value in the “Alias” and “Started on” columns (line 1). Using the agg function, we can decide for each column which value to use for the new row that replaces all the rows corresponding to the same attempt. For example, we can use the value of the lowest column (line 2). However, it is also possible to use the average of all the values or a concatenation of them. The latter method is used in line 5 for the “Response 1” column: All values in this column are concatenated using the separator <NEW ROW>. Once this procedure is complete, the new rows can be reduced to one attempt per student. At the latest, statistical data evaluation and data visualisation are possible. Data on the total number of attempts for each student and the average number of attempts per student are examples.

The approach discussed in this paper has some advantages, with the possibility of matching the data with other data such as surveys leading the way. Nevertheless, there are other approaches, such as the graphical visualisation introduced by [LEP21]. And for many cases the STACK basic question use report is a sufficient tool. Overall, the choice of tool and approach for analysing student responses to STACK questions depends on your needs and purposes.

Bibliography

- [ALK19] Ademi, Neslihan; Loshkovska, Suzana; Kalajdziski, Slobodan: Prediction of Student Success Through Analysis of Moodle Logs: Case Study. In (Gievska, Sonja; Madjarov, Gjorgji; Prediction of Student Success Through; Analysis of Moodle Logs: Case Study, eds): ICT Innovations 2019. Big Data Processing and Mining: 11th International Conference, ICT Innovations 2019, Ohrid, North Macedonia, October 17–19, 2019, Proceedings, volume 1110 of Communications in Computer and Information Science, pp. 27–40. Springer International Publishing, Cham, 2019.
- [LEP21] Landenfeld, Karin; Eckhoff, Malte; Priebe, Jonas: Graphical Visualization of STACK Response Analysis. In: Contributions to the International Meeting of the STACK Community 2021. TTK University of Applied Sciences, Tallinn, Estonia, 2021.
- [LM22] Lache, Jonas A.; Meißner, Daniel: How to STACKrate: The Development of an Intuitive Evaluation Tool. *International Journal of Emerging Technologies in Learning (iJET)*, 17(23):7–14, December 2022.