Midterm Project

Maggie Sha

2020/11/27

Abstract

Riiid Labs is a company that aims to improve the education market by providing AI solutions. This project uses the datasets which contain more than 100 million student interactions from Riiid Labs and aims to fit a suitable model for "Knowledge Tracing", modeling student knowledge over time and to predict whether or not the student could answer the next question correctly. The best model is expected to be used to create personalized learning experiences for all the students with the Internet connection and to improve educational equity.

Introduction

Background

Riiid launched an AI tutor based on deep-learning algorithms in 2017 that attracted more than one million South Korean students. The datasets for this project documented those students' historic performance, the performance of other students on the same question, metadata about the question itself, and more.

Datasets

There are three datasets contains 19 columns. With limited space, I will only show columns that are relevant to understand my model here. The full datasets can be found at https://www.kaggle.com/c/riiid-test-answer-prediction/data

The main dataset is train.csv, some noteworthy columns are:

- timestamp: (int64) the time in milliseconds between this user interaction and the first event completion from that user.
- user_id: (int32) ID code for the user.
- content_id: (int16) ID code for the user interaction
- content_type_id: (int8) 0 if the event was a question being posed to the user, 1 if the event was the user watching a lecture.
- answered_correctly: (int8) if the user responded correctly. Read -1 as null, for lectures.
- prior_question_elapsed_time: (float32) The average time in milliseconds it took a user to answer each question in the previous question bundle.
- prior_question_had_explanation: (bool) Whether or not the user saw an explanation and the correct response(s) after answering the previous question bundle.

Two other datasets question.csv and lecture.csv contain information about questions and lectures. The used column is 'tags', described as one or more detailed tag codes for the question. The meaning of the tags is not provided.

Exploratory Data Analysis

Since train.csv contains more than 100 million data and makes it hard to operate, I use a 2 million subset instead for the EDA. What I have done for the exploratory data analysis are: (1) check and clean NAs (2) do calculation and plot to find potential predictors (3) create new predictors based on the work (4) put all predictors and outcome into one dataset (5) converting grouping variables into factors.

The following text or graphs are helpful to understand my model, all other results from EDA are in the appendix.

User-Level Accuracy and Question-Level Accuracy

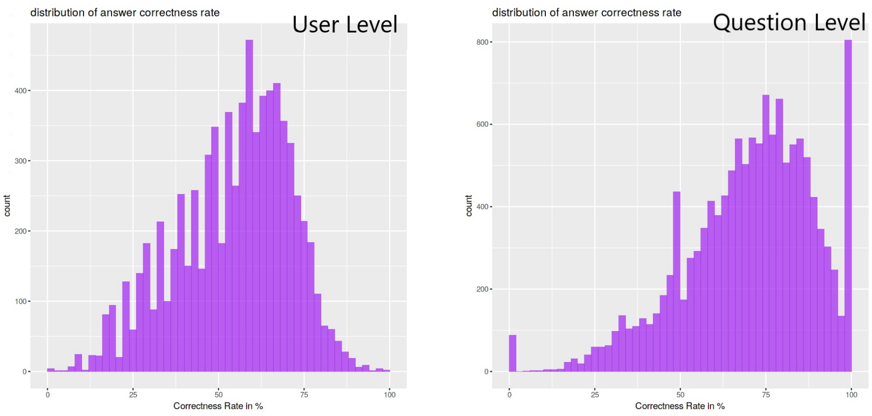
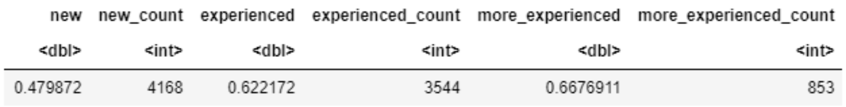


Figure 1: (Distribution of accuracy)

User accuracy is correct questions count divided by total questions count that each user has answered. I find that most of the users have accuracy at around 65%. Since users have different mean accuracy, I put user-level accuracy into my model as a predictor. I think user accuracy would be more reliable if it is together with the total questions count that each user answered, so I also kept question count as a predictor.

Question accuracy is the correct count on each question divided by the total count that each question has been answered. For a similar reason as user accuracy, I use question-level accuracy and the total count of each question as predictors.

Then I want to see whether or not answering more questions would increase their accuracy. I separated the users into three groups based on the number of questions they have answered: 'new' if question count < 50, 'experienced' if question count >= 50, 'more_experienced' if question count >= 500.



new	new_count	experienced	experienced_count	more_experienced	more_experienced_count
<dbl>	<int>	<dbl>	<int>	<dbl>	<int>
0.479872	4168	0.622172	3544	0.6676911	853

Figure 2: (User accuracy in different groups)

It turns out that the 'new' group has an average accuracy of 48%, 'experienced' is around 62% while 'more_experienced' is about 66.8%. This indicates answering more questions increases user accuracy. However, since there is no big improvement from 'experienced' to 'more_experienced', I only keep 'new' and 'experienced' groups and create a grouping variable called 'experience' to identify them.

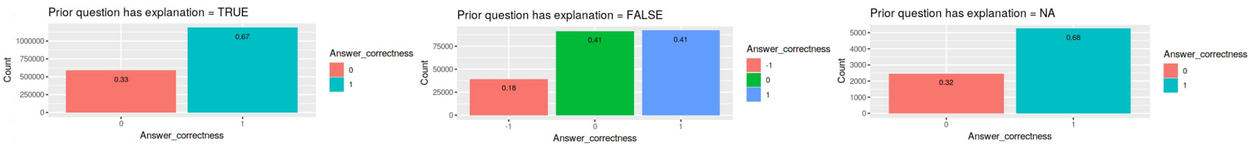


Figure 3: (Explanation)

Does explanation from the prior question affect user accuracy?

Not all questions have a followed explanation after users answering them. I am curious about whether this explanation would affect user accuracy. The result shows the "TRUE" group and "NA" group(which are the first-time users) have a close accuracy, while the "FALSE" group only has 50% accuracy(ignoring -1, 0 and 1 have the same count). Because of this difference, I use prior_question_had_explanation as a predictor.

Does lecture experience affect user accuracy?

```
Never_Attend_Lecture   Lectured
               <dbl>      <dbl>
           0.5046371  0.6108842
```

```
Less_Than_50_Lecture  More_lecture
               <dbl>         <dbl>
           0.5421855     0.6639759
```

Figure 4: (Explanation Result)

As content_type_id indicated, there are two kinds of interactions: Lecture and Question. Although users do not answer questions for the lecture interactions, I still want to confirm the effect of attending lectures on user accuracy. I separate users into 'Never_Attend_Lecture' and 'Lectured' groups and find they have different average accuracy, 50% for students who never attend any lecture and 61% for students who attended at least one lecture. For this reason, I create a 'lec' grouping variable and use it in my model. I also find students with at least 50 lecture experiences have higher accuracy than students with lower than 50 lecture experience. Thus, I also include the lecture count of each student as a predictor.

Hot Tags

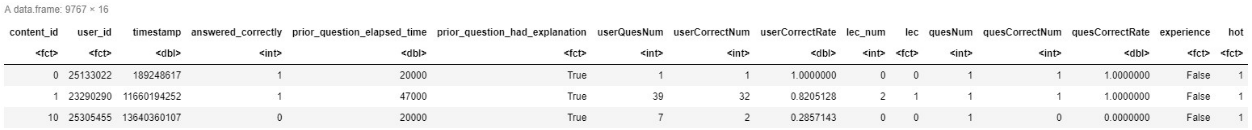
Each question is associated with one or more tags. Considering that I may want to use the multilevel model with content_id as the random effect, the tags can work as descriptions of questions and could improve my model. So I also build a grouping variable "hot" to indicate each question is with the top 20 tags or not.

Method

My model should be a logistic regression model, I decided on using Bayesian modeling by Stan since it is more reliable. I tried both the Bayesian logistic regression model and the Bayesian multilevel logistic regression model with different ways to expand the model. Then I compared those models by looking at their residual plots and results from leave-one-out-cross-validation (loo).

This is my final dataset. As you can see, there are some correlated columns, such as userQuesNum, userCorrectNum, and userCorrectRate. Since I am using Bayesian models and the correlation is not too large, it should be moderate to put them into the model. To see the difference, I use two different ways to expand the same model.

The models I fitted are:



A data.frame: 9767 × 16

content_id	user_id	timestamp	answered_correctly	prior_question_elapsed_time	prior_question_had_explanation	userQuesNum	userCorrectNum	userCorrectRate	lec_num	lec	quesNum	quesCorrectNum	quesCorrectRate	experience	hot
<fct>	<fct>	<dbl>	<int>	<dbl>	<fct>	<int>	<int>	<dbl>	<int>	<fct>	<int>	<int>	<dbl>	<fct>	<fct>
0	25133022	189248617	1	20000	True	1	1	1.0000000	0	0	1	1	1.0000000	False	1
1	23290290	11660194252	1	47000	True	39	32	0.8205128	2	1	1	1	1.0000000	False	1
10	25305455	13640360107	0	20000	True	7	2	0.2857143	0	0	1	0	0.0000000	False	1

Figure 5: (Dataset)

- The first model is a Bayesian binary logistic regression model (stan_glm (family = 'binomial')) using all columns except 'user_ID' and 'content_ID' as predictors on 'answered_correctly'.
- The second model is a Bayesian binary logistic regression model (stan_glm (family = 'binomial')), excluding the correlated columns 'userCorrectNum' and 'quesCorrectNum' and the two ID columns, using others as predictors on 'answered_correctly'.
- The third model is a Bayesian binary multilevel logistic regression model with content_id as the random effect (stan_glmer (family = 'binomial')), using all columns expect 'user_ID' as predictors on 'answered_correctly'.

Result of Residual Plot and Loo Comparison

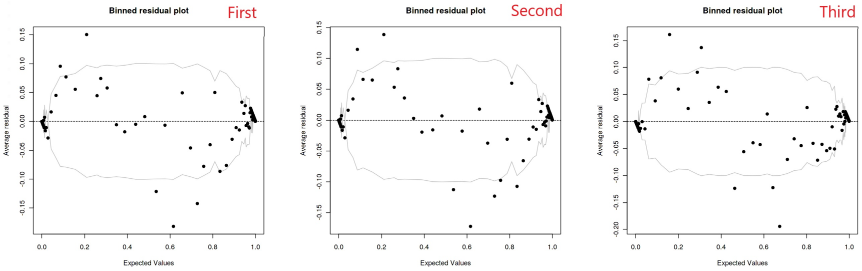
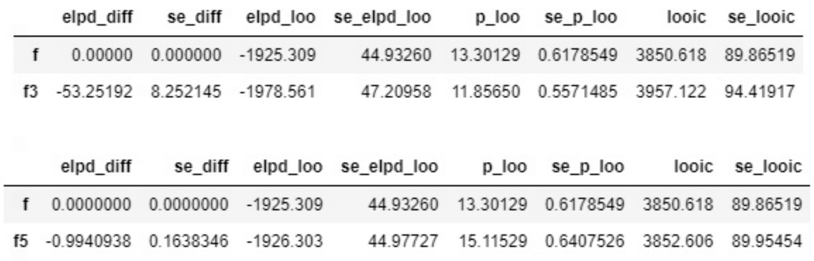


Figure 6: (Residual)

These are the residual plots of the three models.



	elpd_diff	se_diff	elpd_loo	se_elpd_loo	p_loo	se_p_loo	looic	se_looic
f	0.00000	0.000000	-1925.309	44.93260	13.30129	0.6178549	3850.618	89.86519
f3	-53.25192	8.252145	-1978.561	47.20958	11.85650	0.5571485	3957.122	94.41917

	elpd_diff	se_diff	elpd_loo	se_elpd_loo	p_loo	se_p_loo	looic	se_looic
f	0.0000000	0.0000000	-1925.309	44.93260	13.30129	0.6178549	3850.618	89.86519
f5	-0.9940938	0.1638346	-1926.303	44.97727	15.11529	0.6407526	3852.606	89.95454

Figure 7: (Loo1)

On the top is loo1, it is the loo comparison between the first model (f) and the second model (f3).

On the bottom is loo2 and it is the loo comparison between the first model (f) and the third model (f5).

Discussion

Discussion on the Residual Plot

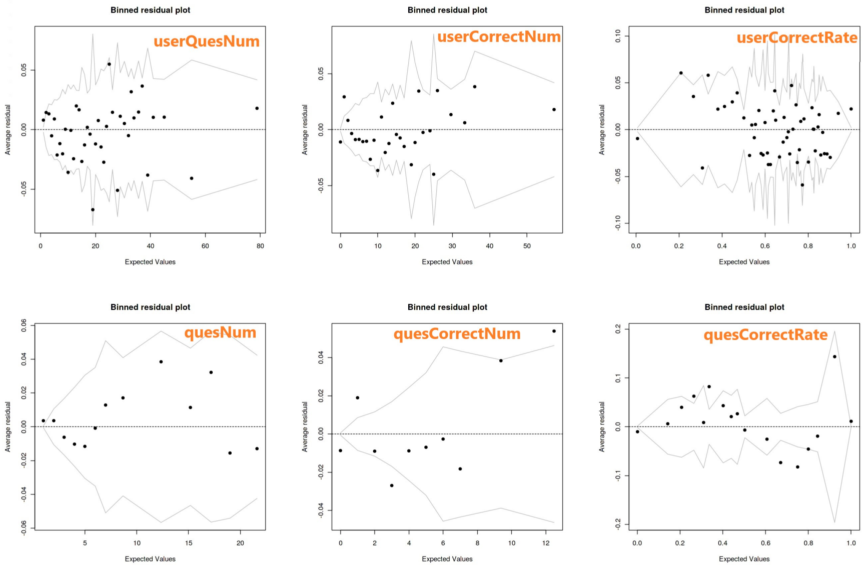


Figure 8: (Residual_on_predictors)

From the residual plots in the Result part, I notice that those three models all perform well on estimating 0 and 1. However, they all show a decreasing trend. To better understand why I assessed the fit on the predictors by plotting the binned residual plot on each predictor. Figure 9 are the plots that indicate some problems. I found all binned residual plots on these six predictors have a systematic pattern, either going downward or upward. I think this is a signal that those predictors have a nonlinear relationship with my outcome variable.

Discussion on the Loo Comparison Result

From the elpd_diff of Loo1, we can see the first model is much better than the second model, and Loo2 shows the first model is slightly better than the third model, but there is no big improvement. So the loo comparison result suggests my best fit so far is the first model.

Why the binary logistic regression model is doing better than the multilevel model? One reason in my mind is that we are using content_id as the random effect, but as I talked before, those question-related predictors are not predicting our outcome as expected, so before I figure out how should I modify them and put the new terms into the models, multilevel regression model will not make any difference.

Next Steps

- Research on how to include the nonlinear terms in the logistic regression model.
- Include nonlinear terms for those six predictors 'userQuesNum', 'userCorrectNum', 'userCorrectRate', 'quesNum', 'quesCorrectNum', and 'quesCorrectRate' in both binary logistic model and multilevel binary logistic model and see does it make any improvement.
- Maybe there are other relevant independent variables. I will continue looking for the potential new predictor.

Acknowledgements

I would like to thank my professor Masanao Yajima and TA Tim Hogan for answering all my questions and giving me suggestions. They helped me understand my work better.

Appendix

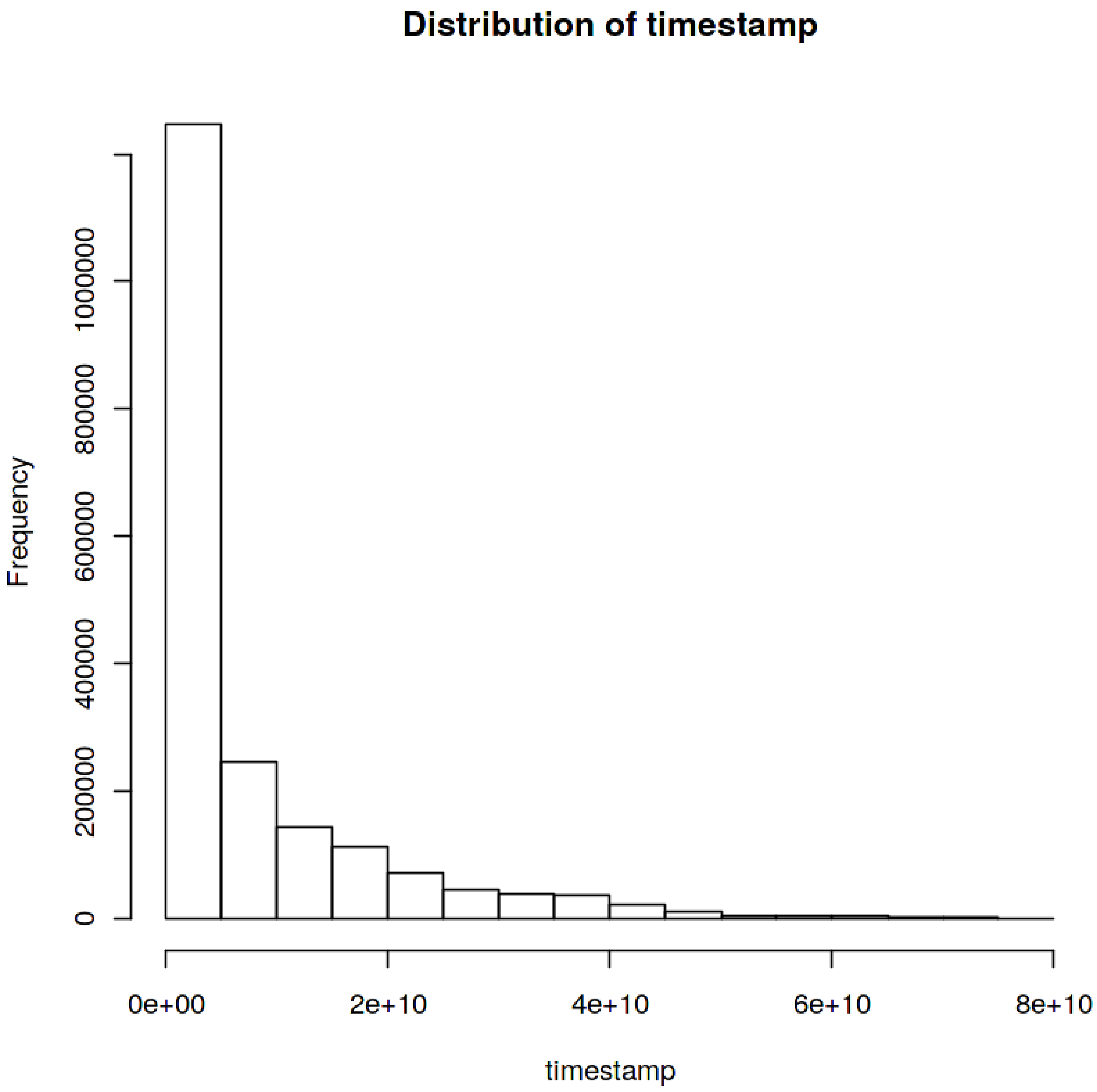


Figure 9: (Residual_on_predictors)

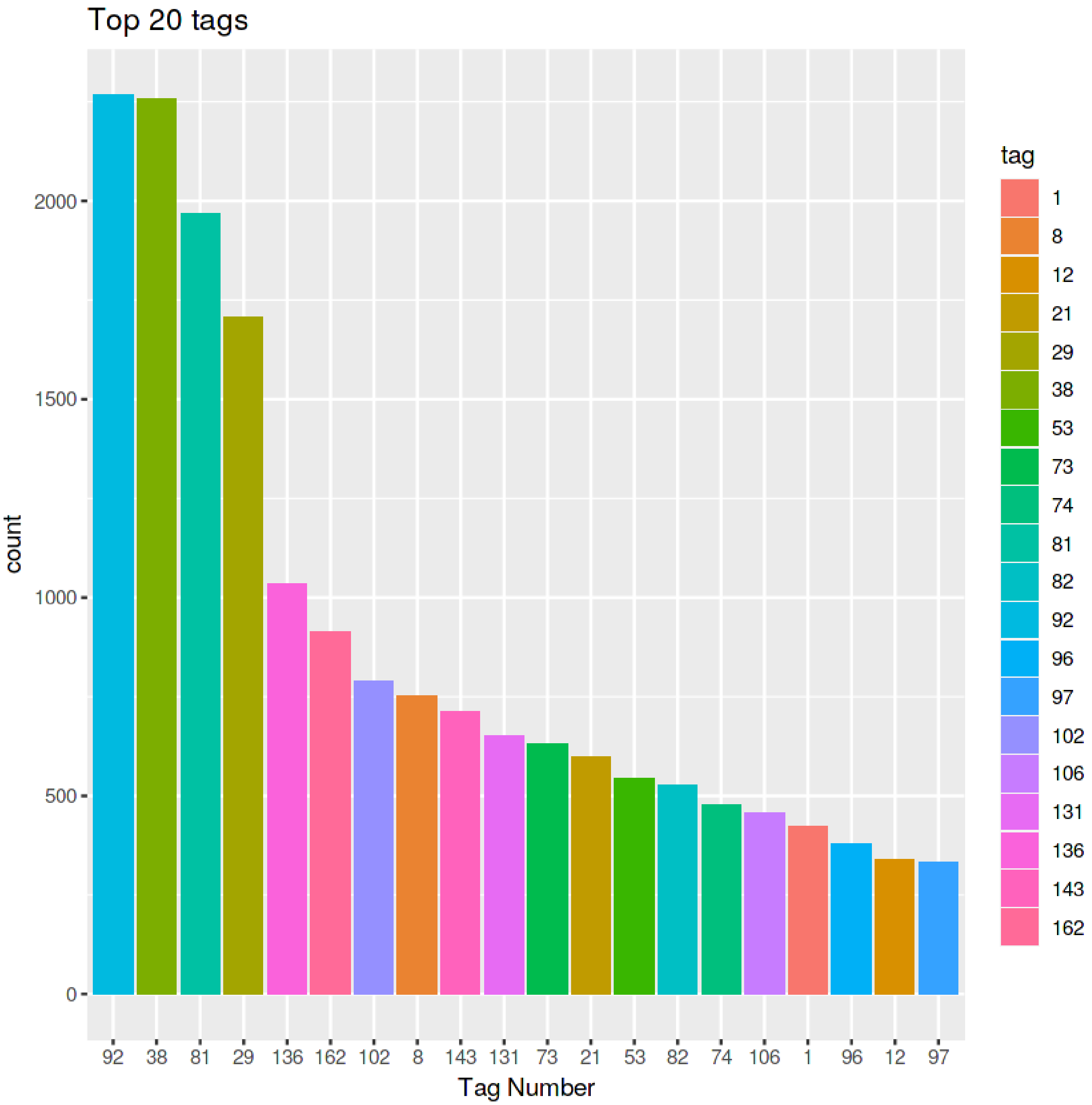


Figure 10: (Residual_on_predictors)

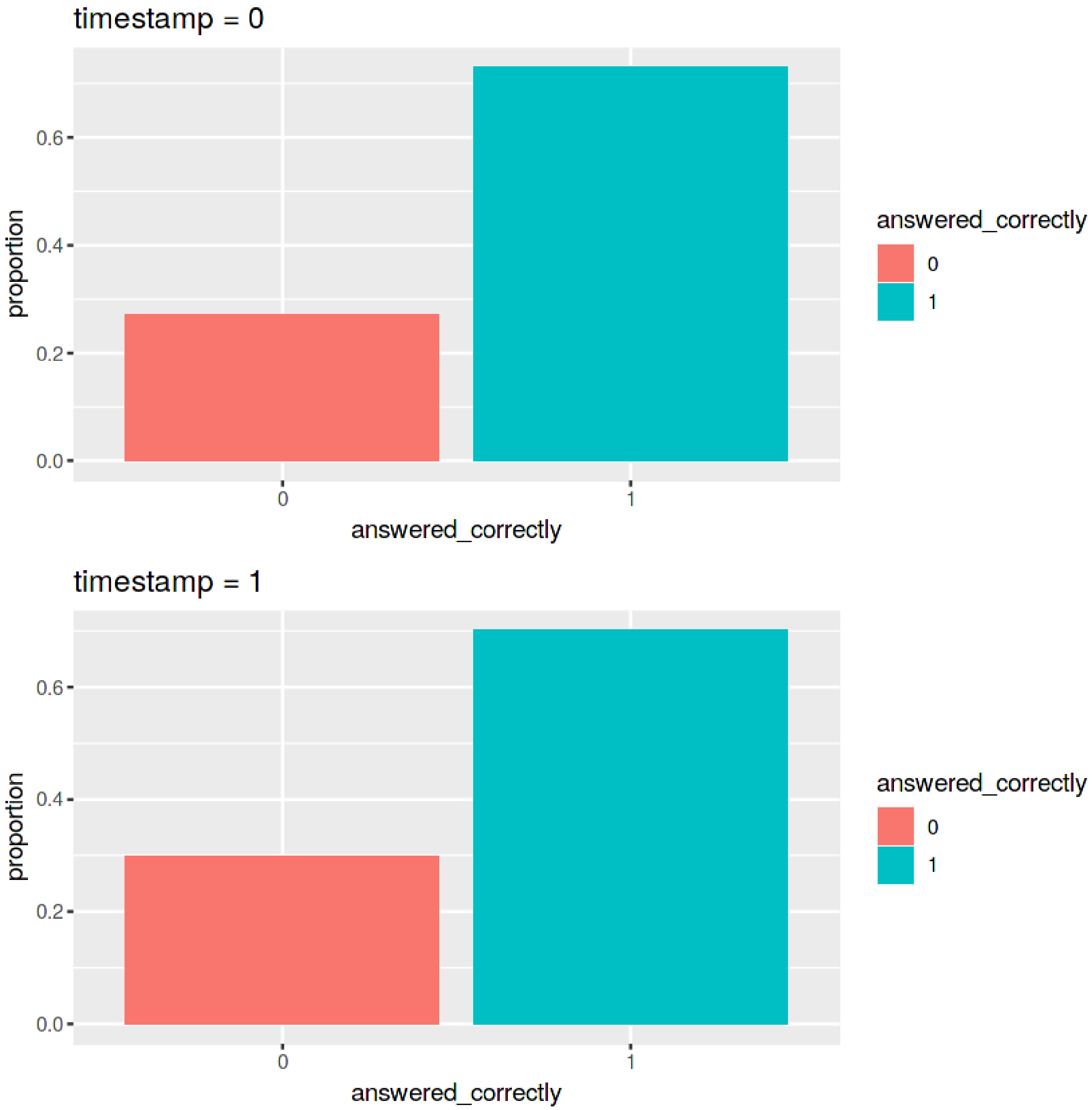


Figure 11: (Residual_on_predictors)

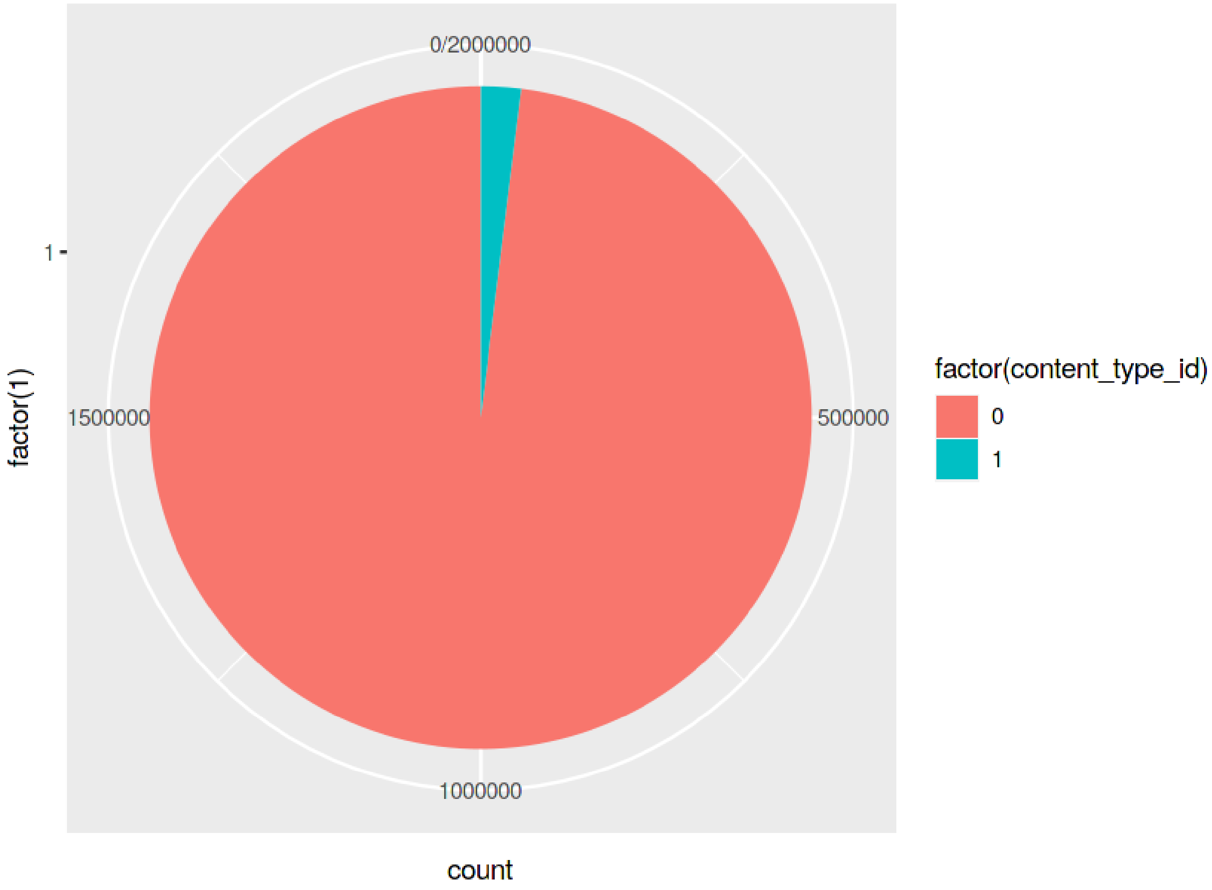


Figure 12: (Residual_on_predictors)

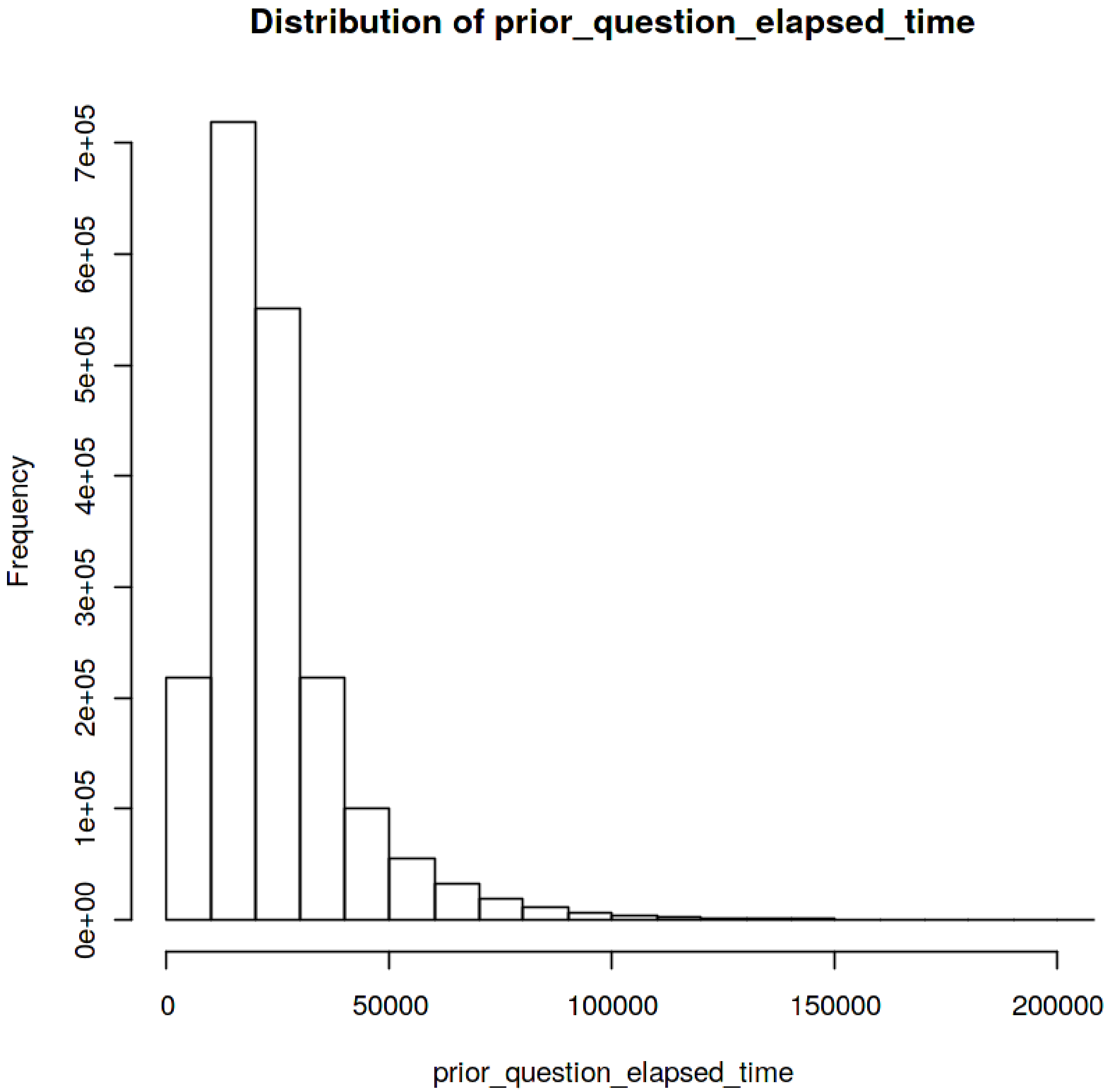


Figure 13: (Residual_on_predictors)

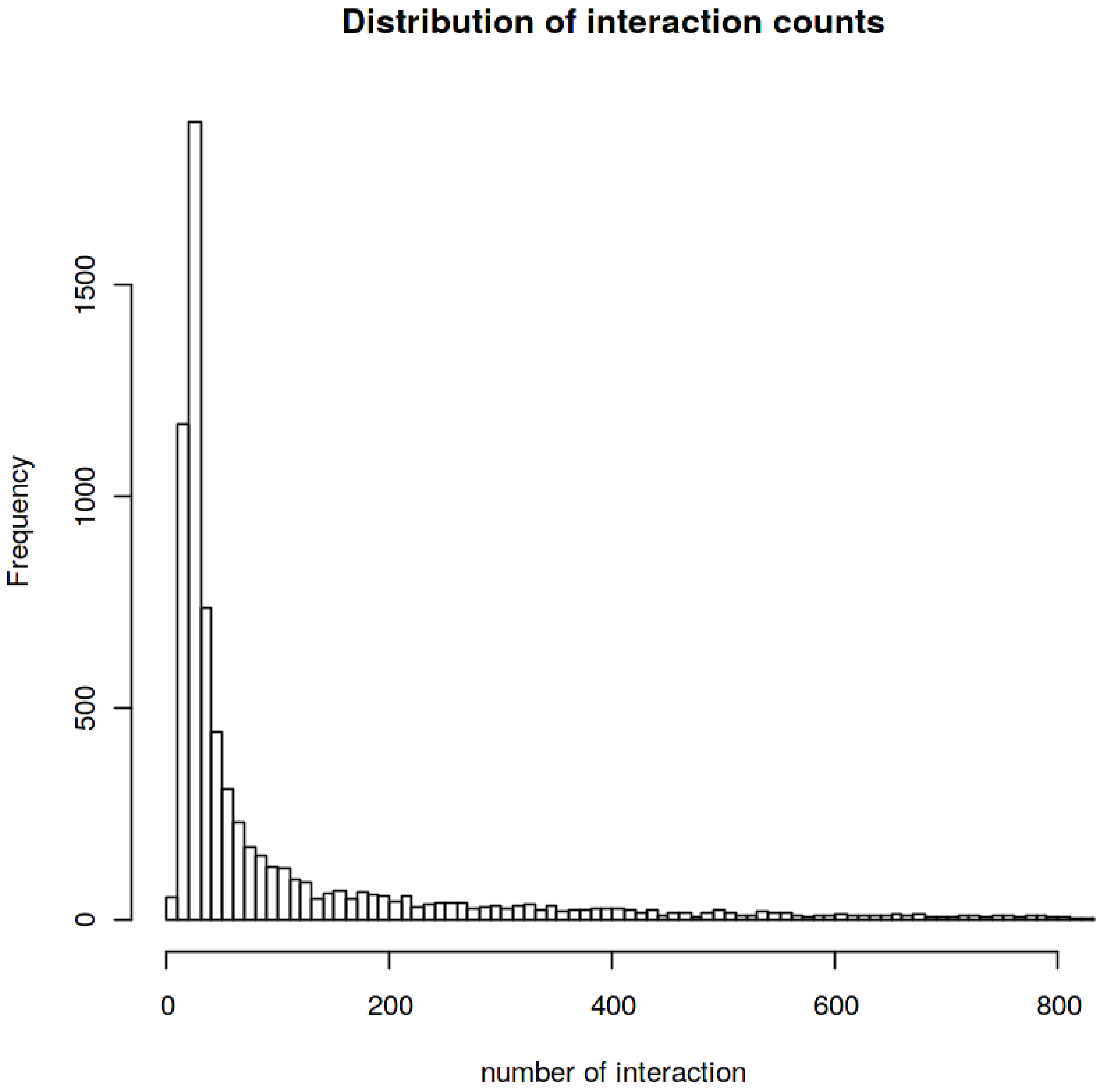


Figure 14: (Residual_on_predictors)

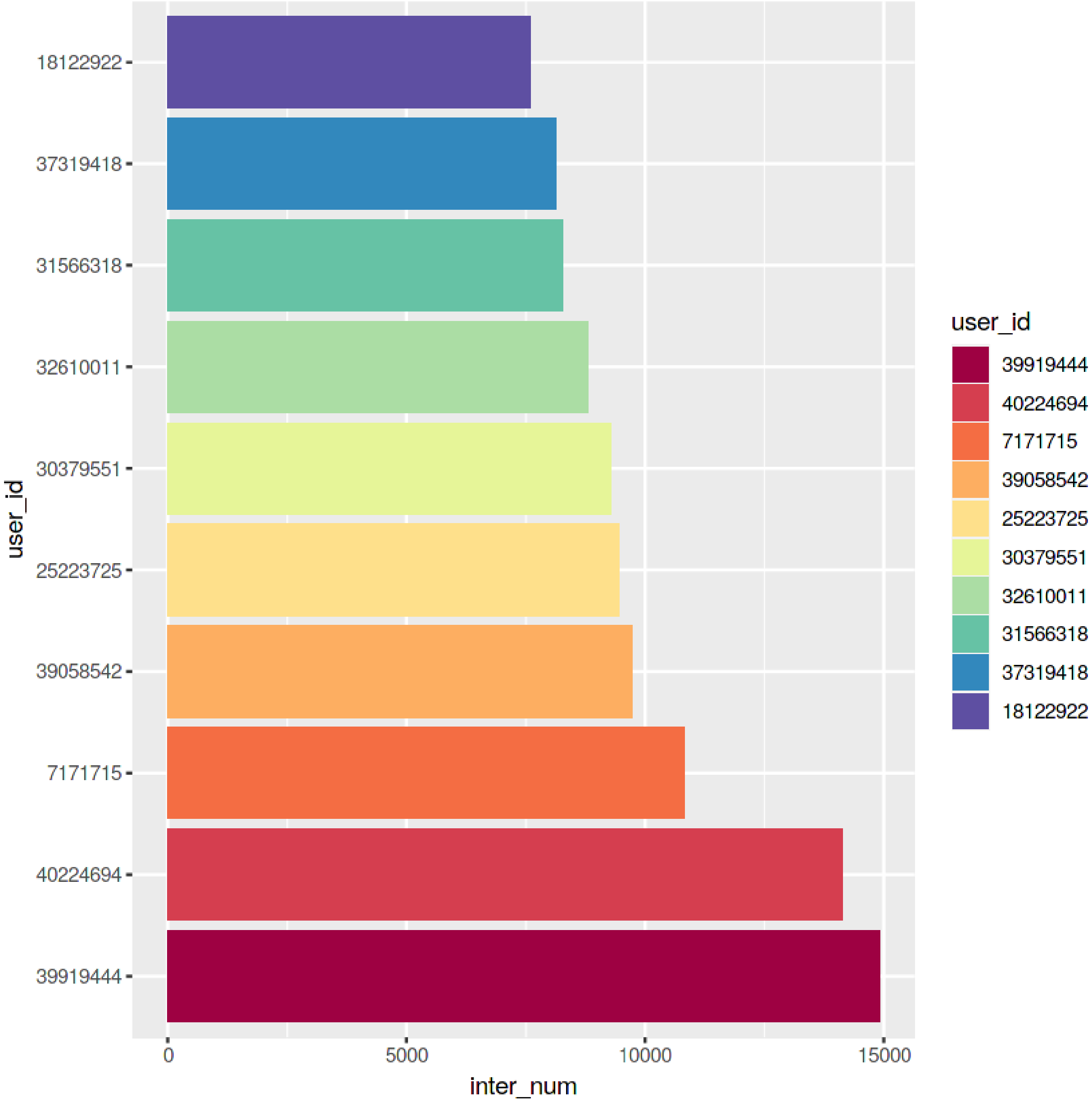


Figure 15: (Residual_on_predictors)

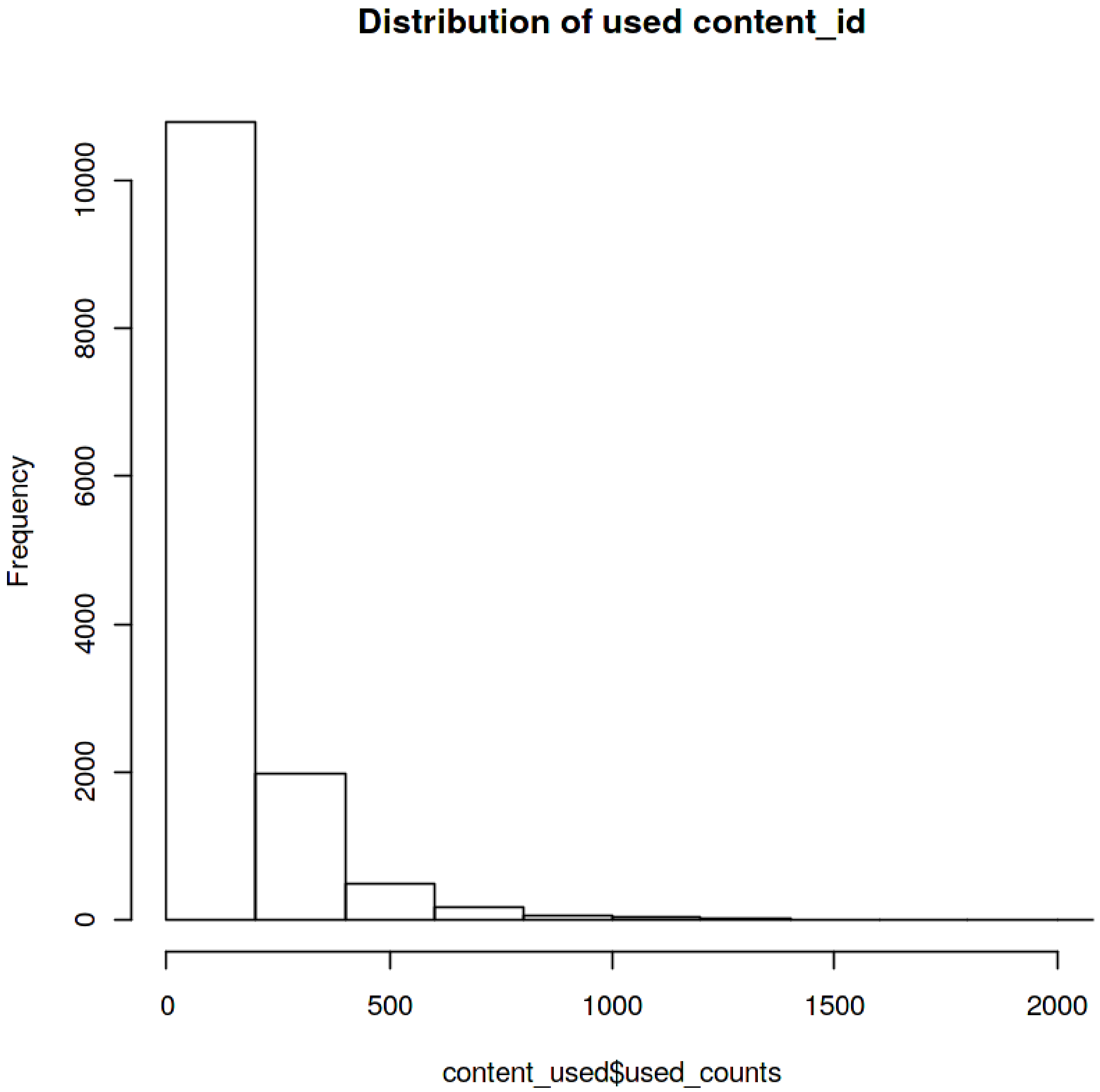


Figure 16: (Residual_on_predictors)

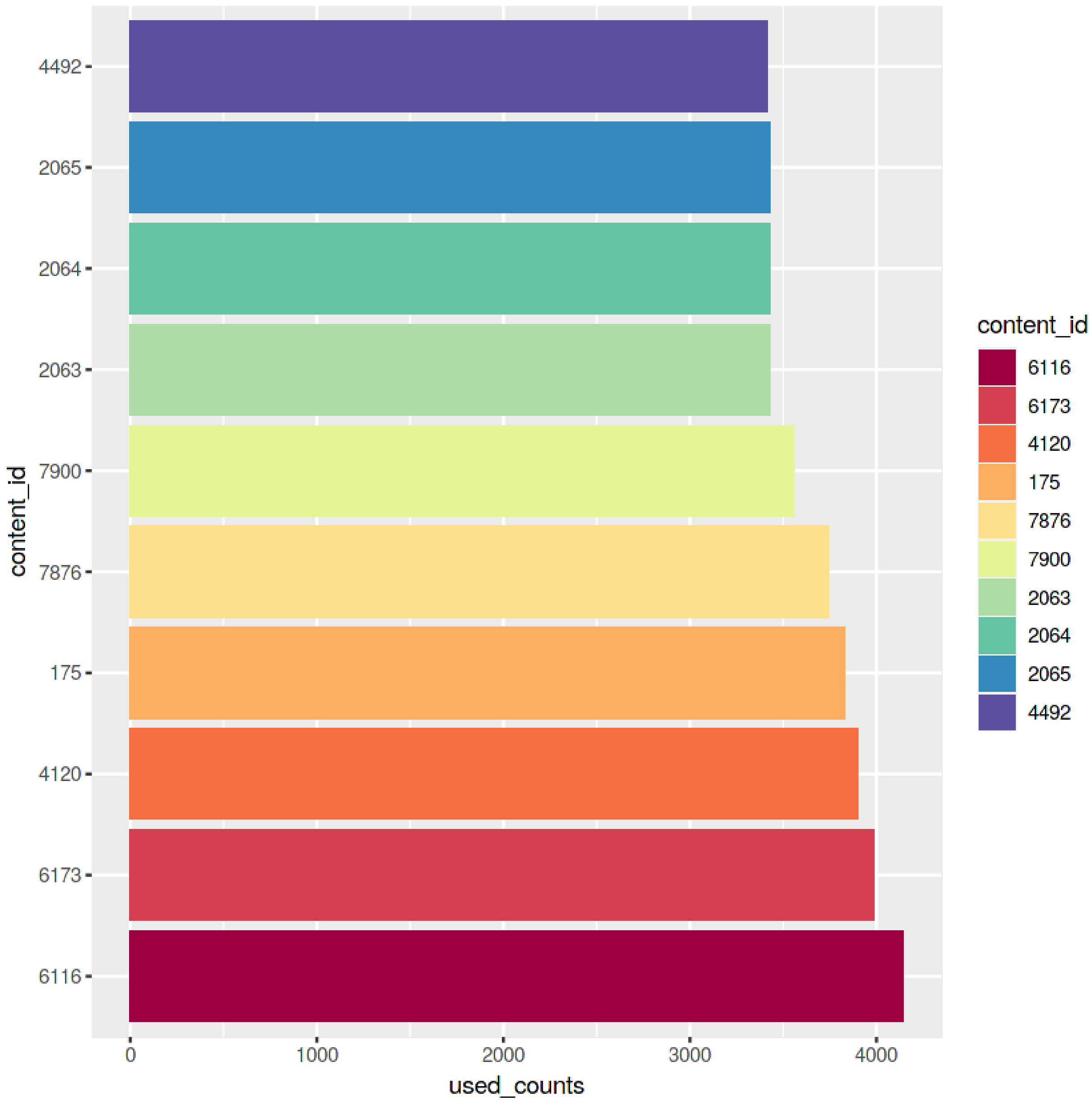


Figure 17: (Residual_on_predictors)

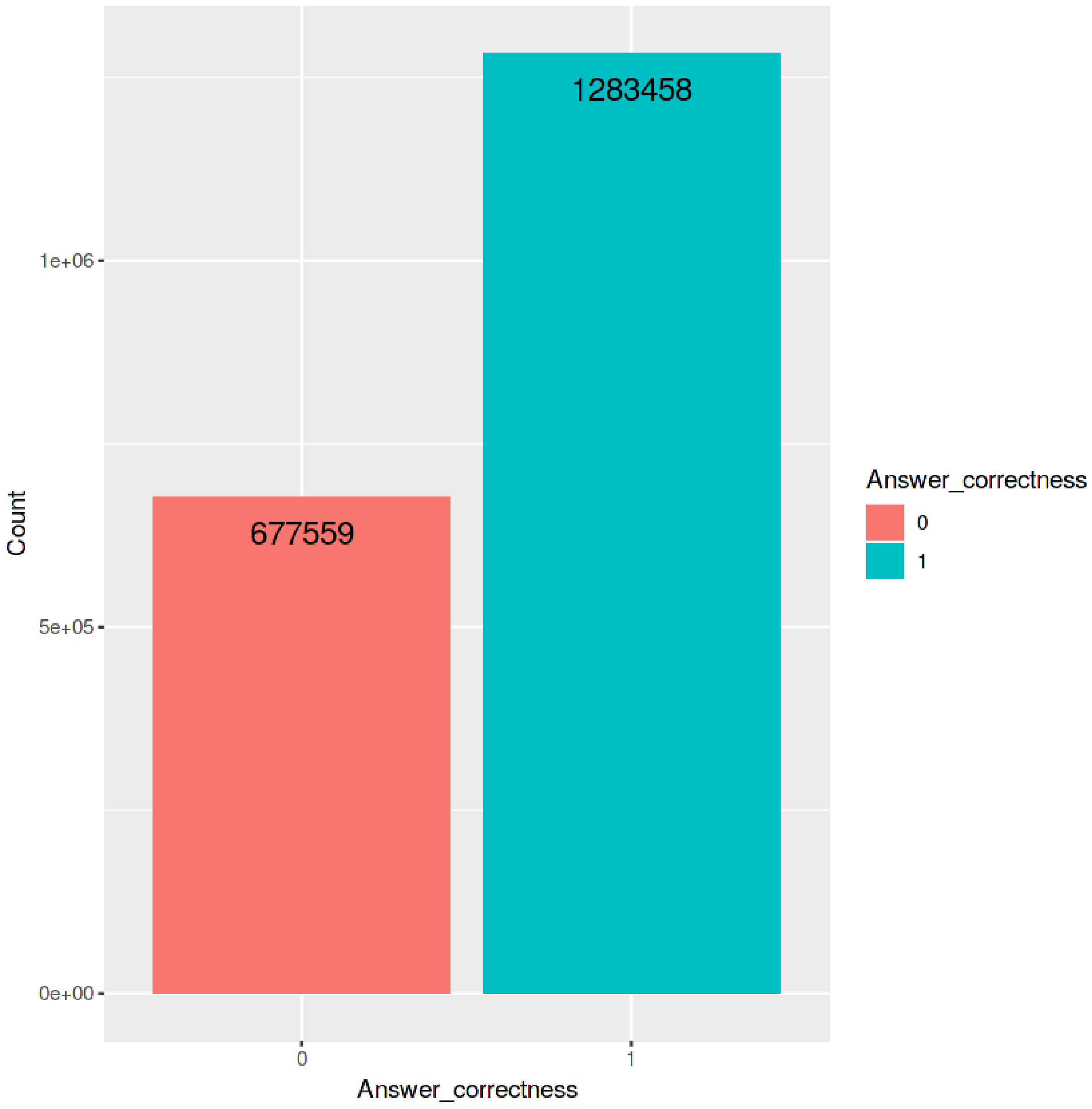


Figure 18: (Residual_on_predictors)

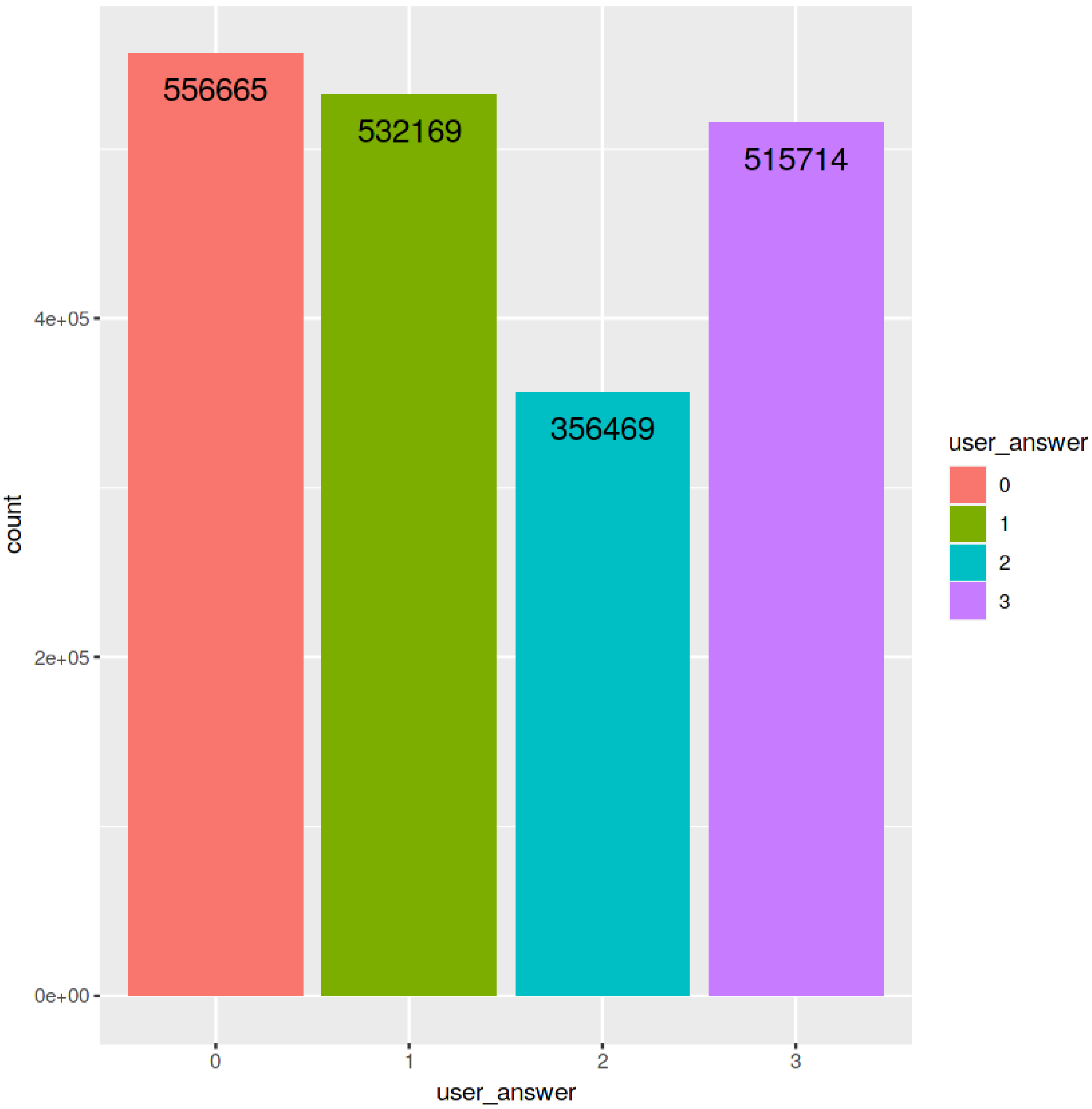


Figure 19: (Residual_on_predictors)

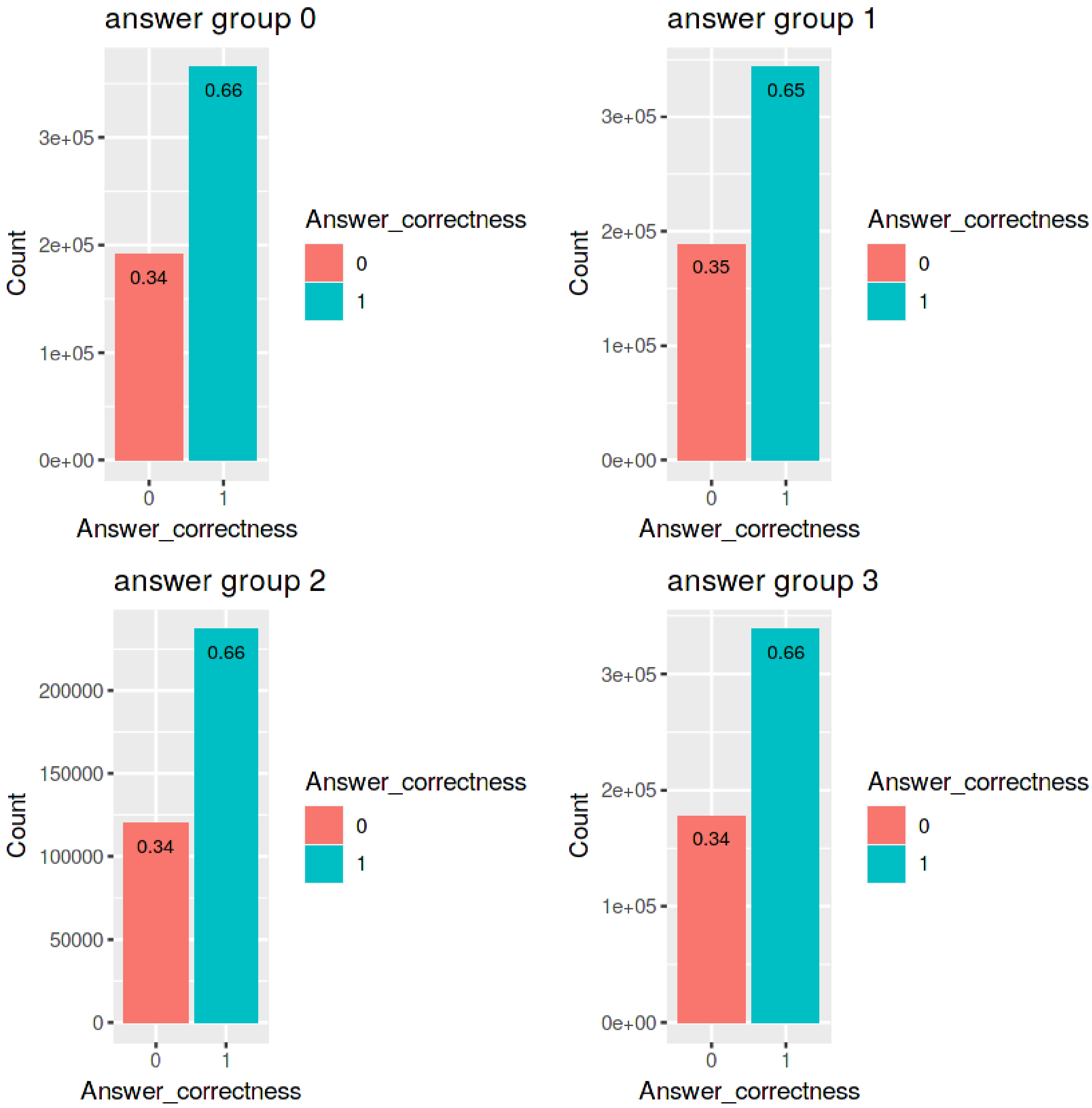


Figure 20: (Residual_on_predictors)

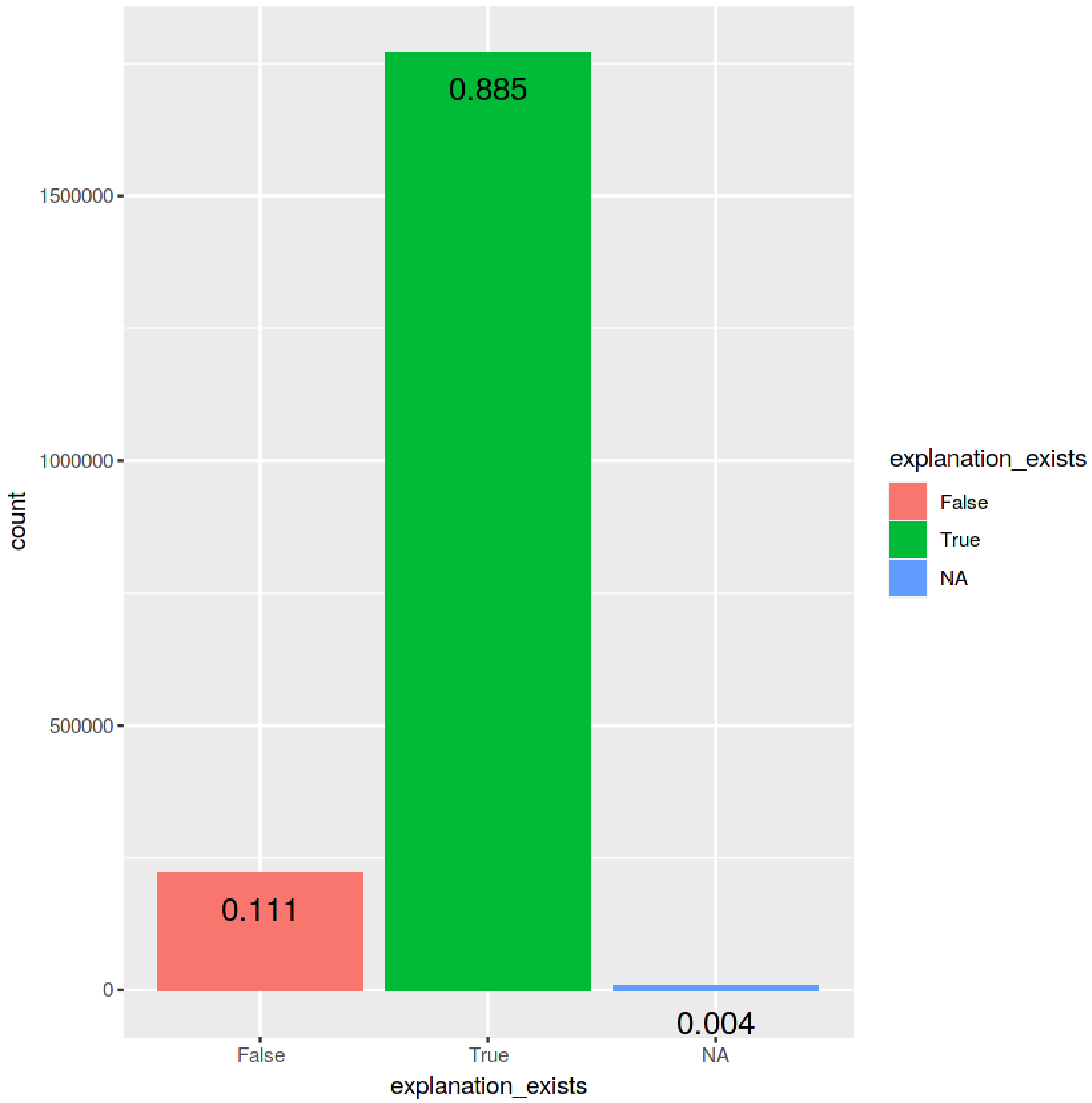


Figure 21: (Residual_on_predictors)

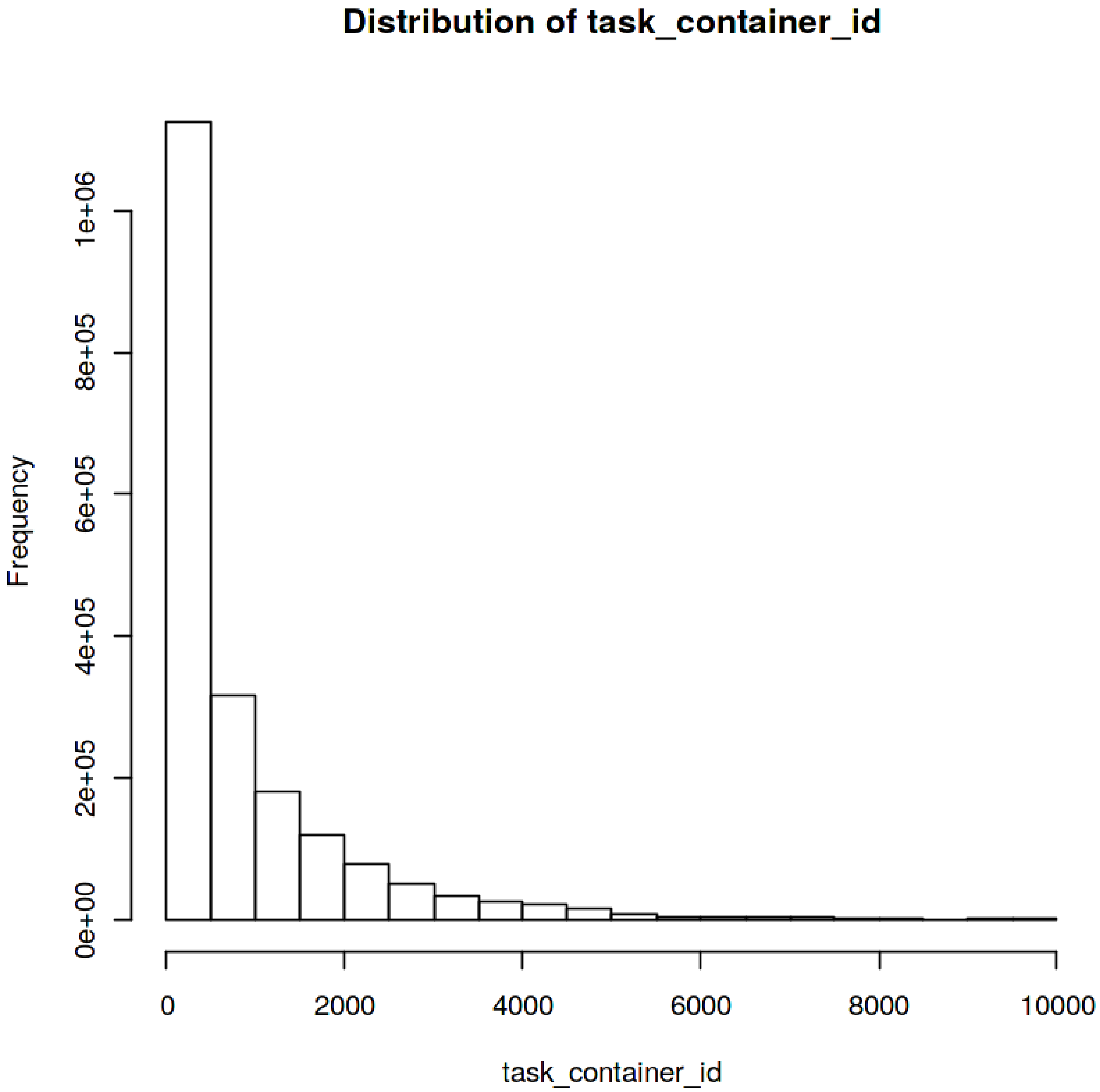


Figure 22: (Residual_on_predictors)

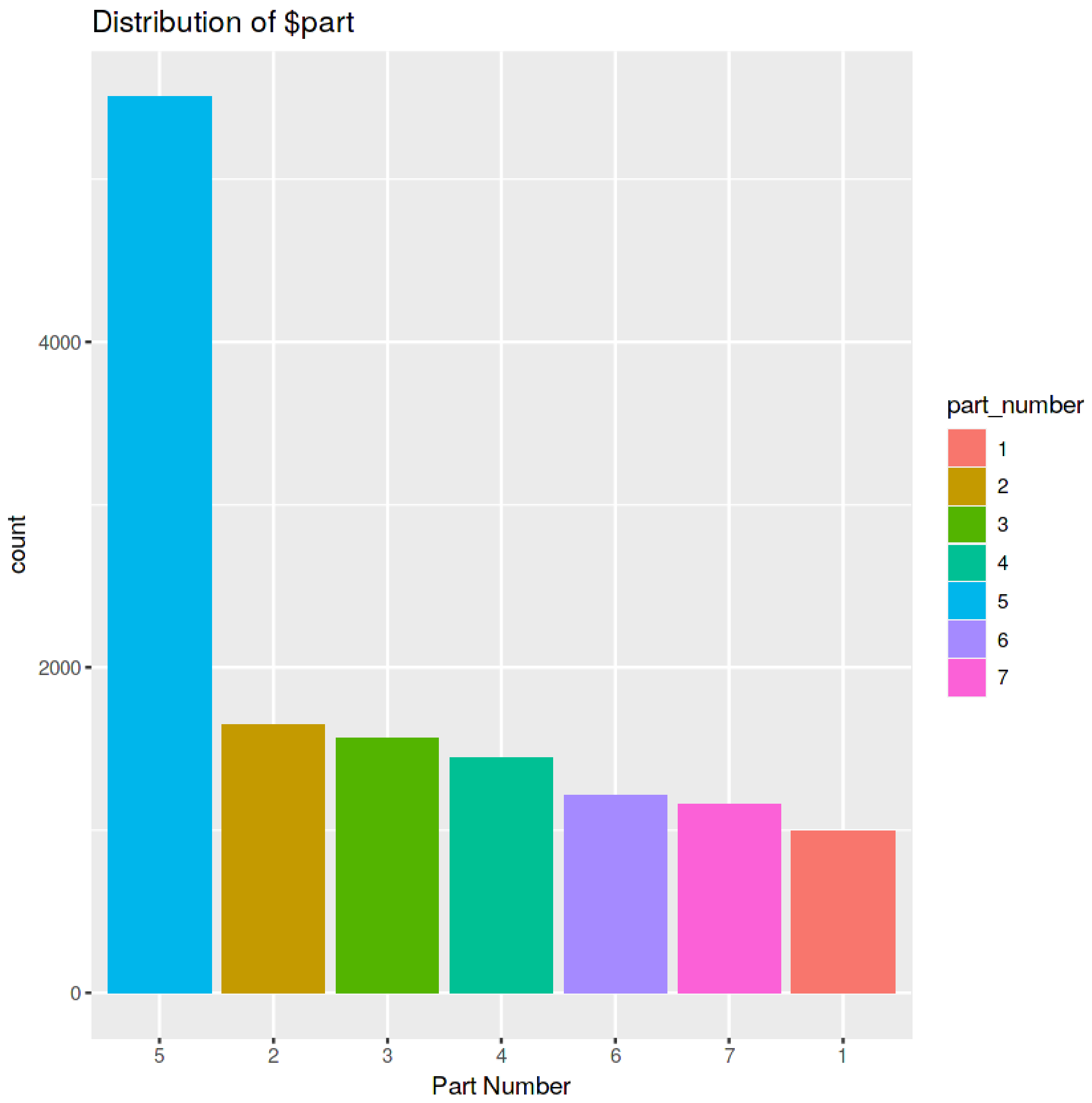


Figure 23: (Residual_on_predictors)

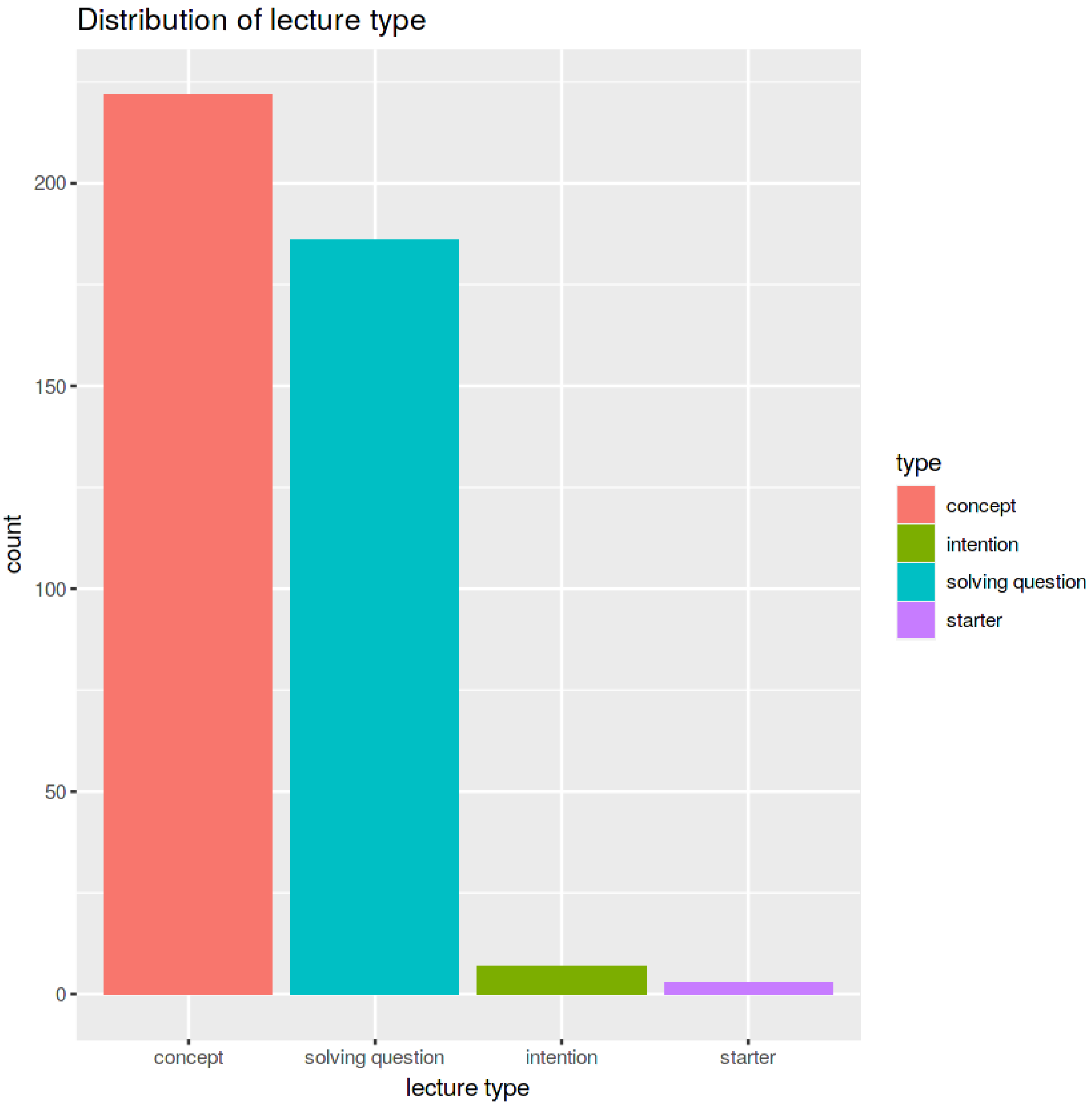


Figure 24: (Residual_on_predictors)