# Traits of accepted answers in Mathematics and MathOverflow


## ABSTRACT

MathOverflow and Mathematics were started in 2009 and 2010 respectively mentioned [1] along with [8] and became a forum where mathematicians collaborated and communicated to solve new and existing math problems. For each query posted, a questioner has the option of marking an answer as the best answer (or accepted answer) which informs the other users that the marked answer has helped the questioner solve his/her problem. In this particular paper we evaluate the traits of accepted answers, such as the number of votes on an answer, reputation of the answerer and promptness of reply (or answer speed) are factors that affect acceptance of a solution. Thus, we conclude after conducting data analysis, that approximately 75% and 70% of answers on MathOverflow and Mathematics respectively had their accepted answers as highest voted answers. Furthermore, promptness of reply (reply within a week) and user reputation (reputation higher than ten thousand) had little role in acceptance of an answer. At the end, we convey our concern for validity of the results and how the research can be enhanced in the future.


## 1 INTRODUCTION

This research project utilizes Mathematics and MathOverflow questions to discover the types of traits of accepted answers. An answer could contain defining traits such as reputation of the user, content, size, and promptness of a solution. Furthermore, the quality of an answer may depend on user, content, and thread features which are mentioned in [4]. The motives of the replier sometimes might affect the type of solution provided. The solution provider may have the personality of a tutor, due to which the questioner might not get a straightforward solution rather a reply that allows the individual to discover the answer by himself/herself. This may hinder the acceptance of an answer given by a tutor. Additionally, the preference of the questioner might be promptness of the solution or the reputation of the answerer which in turn may also affect the acceptance of an answer. Retrieving multiple solutions from various sources may convey that the questioner can select the best answer according to his/her preference. This research paper will strive to find out the information on whether there is any factor that may lead to acceptance of an answer for a query.

## 2 MOTIVATION

There are mainly two types of users on Mathematics and MathOverflow one which post the questions and have the ability to select an optimal answer to their question. The other users answer questions or search for questions that are already answered and can up or down vote an answer. The questioner generally searches for answers that are easy to interpret and solves his/her problem quickly and marks that answer as accepted solution. However, sometimes the questioner may get perplexed due to high up votes on a complex answer and might select that as an accepted answer. Therefore, those looking for answers might assume an accepted answer to be the correct answer or an optimal answer which may lead to selection of wrong solutions or solutions that are not elaborate or solutions that restrict the users to one method. Furthermore, unlike StackOverflow which mainly contains coding solutions which can be tested using a programming language, the solutions on Mathematics and MathOverflow cannot be tested or verified until and unless the answerer has given the source of the solution. The research question related to high up votes is important in this case as it helps us to understand the thought process of a questioner and those that seek solutions to an already answered question.

As a questioner receives multiple solutions to a particular problem this means that there must be a method to select the best answer. Reputation of the responder, number of comments under an answer and promptness of a reply are different parameters which might be important factors to consider. Most of us tend to accept the information provided to us by an experienced and qualified individual such as a Professor rather than an amateur person even if they give similar advice. This thought process could be observed in Mathematics and MathOverflow because the user reputation score is linked to their expertise [Yla R. Tausczik and James W. Pennebaker, 2016]. Furthermore, a rapid reply also affects our acceptance of the answer [Yao Lu et al., 2020]. Hence, the research question related to popularity and rapid reply forces us to find how and how much these parameters affect an individual's decision on accepting an answer.

## 3 BACKGROUND

The Mathematics and MathOverflow questions and answers are being investigated in this research. The Stack Exchange website contains Mathematics and MathOverflow both of which are Community-based Question Answering (CQA) services. There are over 340 thousand and 20 thousand daily visitors on Mathematics and MathOverflow respectively. Furthermore, Mathematics and MathOverflow have 1.4 million questions along with 1.9 million answers and 125 thousand questions along with 163 thousand answers respectively [1]. Users can reply with a solution for each question posted on these CQA services. The questioner has the ability to up or down vote an answer along with the ability to select a best answer which is called an accepted answer that is denoted by a green tick. An accepted answer is an ideal answer according

to the questioner which is in accordance with their requirements. Moreover, questions along with answers both can be voted up or down by users other than the questioner which makes them easier to find, however, other users cannot accept an answer. Each user has a reputation score attached to them which increases due to the high upvotes on their own questions, answers and edits given by other users. The Reputation score increases by 10 for a solution voted up, 5 for a query voted up and 2 when an edit is accepted [2].

## 4 RESEARCH METHOD

The following section will include research questions, data collection and analysis methods.

### 4.1 Research Questions

This research paper will evaluate Mathematics and MathOverflow questions to investigate miscellaneous traits of the answers to these questions. These different traits would be reputation of an individual who provided the solution, the vote count on questions and answers and the promptness of an answer relative to the date on which the question was created. The research questions will be: RQ1: What is the possibility of an accepted answer to be highest voted answer when multiple answers are present? An appropriate hypothesis for RQ1 is that an answer will be accepted if it has highest up-votes compared to other answers. RQ2: Does acceptance of an answer depends on immediate response or reputation of the answerer? An appropriate hypothesis for RQ2 is that answer will be accepted if the responder has a highest reputation and responds to the question promptly.

### 4.2 Data Collection

The main data source that is necessary to evaluate research questions, are the question posts with multiple replies on MathOverflow and Mathematics. A popular question would have numerous solutions that will attract higher number of up-votes and down-votes due to increased probability of people having the same problem. Furthermore, the MathOverflow and Mathematics questions must contain answers that were accepted by the questioner. All this data will help answer RQ1 and RQ2, as it will contain various type of users with different reputations and different timelines in which the questions were answered.

The data is going to be retrieved from Stack Exchange Data Dump at https://archive.org/download/stackexchange website. Out of eight data dumps in Mathematics and seven data dumps in MathOverflow only two important to answer research question. The title of first file is “math.stackexchange.com.7z” and of second file is “mathoverflow.net.7z”. Both of these files contain separate “Posts.xml” file and “Users.xml” file. The “Posts.xml” file contains all posts where each row represents a new post. The “Users.xml” file contains all users where each row represents a new user.

### 4.3 Data Analysis

*4.3.1 Data Curation.* Yla R. Tausczik and James W. Pennebaker [13] used the data from MathOverflow. Similarly, the data will be extracted and stored using Microsoft Power BI, where one table will store the information of “Posts.xml” file and the other table will store entries of “Users.xml”. The “Posts.xml” and “Users.xml” would be loaded in Power BI file with the help of MySQL Workbench. The table will be named as Posts and Users respectively. There would be two “Posts” table, one would be made using MathOverflow “Posts.xml” file and the other would be created using Mathematics “Posts.xml” file. Similarly, two “Users” table would be created using “Users.xml” of both sites.

The Posts table of MathOverflow and Mathematics contains Id, PostTypeId, AcceptedAnswerId, ParentId, CreationDate, DeletionDate, Score, ViewCount, Body, OwnerUserId, Owner-DisplayName, LastEditorUserId, LastEditorDisplayName, Last-EditDate, LastActivityDate, Title, Tags, AnswerCount, Comment-Count, FavoriteCount, ClosedDate, and CommunityOwnedDate headers. Observing the Posts table, the only useful table headings were Id, PostTypeId, AcceptedAnswerId, ParentId, CreationDate, Score, ViewCount, OwnerUserId and AnswerCount. When the value of PostTypeId is 1 then it is question posted by the user and value of 2 under PostTypeId represents an answer. Therefore, the Posts Table will be further divided into the Questions table and Answers table which will further be converted into Questions.xlsx file and Answers.xlsx file respectively using Microsoft Power BI. The Questions table will be created by following useful headers Id, PostTypeId, AcceptedAnswerId, ParentId, CreationDate, Score, ViewCount, OwnerUserId and AnswerCount found in the Posts table along with value of PostTypeId to be 1, AcceptedAnswerId not to be NULL, Score to be greater than 0 and AnswerCount to be greater than 1. The Answers table will have the same headers as Questions table, however, there is only one filter that is PostTypeId is equal to 2. Thus, the final files would be Questions.xlsx and Answers.xlsx.

The Users table is converted to Users.xlsx and contains Id, Reputation, CreationDate, DisplayName, LastAccessDate, WebsiteUrl, Location, AboutMe, Views, UpVotes, DownVotes, ProfileImageUrl, EmailHash and AccountId headers. Observing the Users table, the only useful table headings were Id and Reputation. The final Users table will only contain Id and Reputation column and will be renamed as Profiles and will be converted to Profiles.xlsx using Microsoft Power BI.

*4.3.2 Sampling.* There were over two hundred thousand (215324 to be exact) relevant questions (these were in Questions.xlsx file) in Mathematics and twenty thousand (20650 to be exact) relevant questions in MathOverflow that can be examined to answer RQ1 and RQ2. Furthermore, there were over one million (1859015 to be exact) relevant answers (these were in Answers.xlsx) in Mathematics and one hundred thousand (160485 to be exact) relevant answers in MathOverflow. As the whole population is known and the datasets of relevant questions and answers contain large amount of data. Therefore, it would be time efficient to sample the datasets of relevant answers and questions. After utilizing a type of probability sampling known as Simplified Random Sampling (SRS) 1000 samples were retrieved each for Mathematics and MathOverflow respectively. Moreover, each

sample element out of 1000 samples has a probability of 1/215324 and 1/20650 for being selected for Mathematics and MathOverflow respectively. These 1000 samples of relevant questions of Mathematics and MathOverflow each were stored in a separate file with sampleQuestions.xlsx name. Then ParentId header values in the Answers.xlsx file were compared with the ParentId header values in sample-Questions.xlsx and the entire row of Answers.xlsx was stored in sampleAnswers.xlsx for the ParentId values that matched in sampleQuestions.xlsx. Similarly, AcceptedAnswerId in sample-Questions.xlsx was compared with Id in sampleAnswers.xlsx and when AcceptedAnswerId and Id values matched, then the sampleAnswers.xlsx entry was stored in sampleAnswersAA.xlsx. Then Id in sampleAnswersAA.xlsx was compared with Id in sampleAnswers.xlsx and the common values of Id between sampleAnswersAA.xlsx and sampleAnswers.xlsx are deleted from sampleAnswers.xlsx (this is called Left Anti in Power BI) and stored in sampleAnswersNA.xlsx. The Owner-UserId in sampleAnswersAA.xlsx and sampleAnswersNA.xlsx was compared with Id in Profiles.xlsx and stored in sample-ProfilesAA.xlsx and sampleProfilesNA.xlsx respectively. The sampleProfilesAA.xlsx contains Id and Reputation of accepted answers and sampleProfilesNA.xlsx contains Id and Reputation of non-accepted answers.

The SRS sampling was achieved by using Power BI's Power Query tool. First, sampleQuestions.xlsx file was selected and then a custom column was added in the file by a command (Number.Random()) and then the custom column was sorted, then top 1000 values were extracted from the sampleQuestions.xlsx file. Then Merge Queries tool was used to match ParentId in Answers.xlsx and Id in sampleQuestions.xlsx to get sample-Answers.xlsx. Furthermore, Merge Queries tool was used to match Id in Profiles.xlsx and OwnerUserId in sampleQuestions.xlsx to get samplsProfiles.xlsx.

After performing SRS sampling, four datasets were generated that are sampleQuestions.xlsx, sampleAnswers.xlsx and sampleProfilesAA.xlsx and sampleProfilesNA.xlsx. There were multiple answers in sampleAnswers.xlsx to each question in sampleQuestions.xlsx of a user in sampleProfilesAA.xlsx and sampleProfilesNA.xlsx.

*4.3.3 Statistical Analysis* To answer RQ1, several commands in Python would be utilized to create a AnswerId DataFrame containing Id's of answers with highest Score present in sampleAnswers.xlsx file. To extract these answer Id's, each question Id in sampleQuestions.xlsx was compared with all ParentIds in sampleAnswers.xlsx. This creates a DataFrame containing several answer Ids and out of these answer Ids one with the maximum Score was selected and stored in a final AnswerId DataFrame. At last, the Answer Id DataFrame was compared with AcceptedAnswerId column in sampleQuestions.xlsx and for each matching value, a counter's value was increased by 1 and finally the percentage of highest voted answers that were accepted is calculated by dividing counter value by 1000.

In order to answer RQ2, we will utilize the sampleQuestions.xlsx and sampleAnswers.xlsx to find out the time difference between the posting of a question (under CreationDate header in sampleQuestions.xlsx) and its answer being delivered (under CreationDate header in sampleAnswers.xlsx) and store the time difference in the column SpeedM in file sampleAnswers.xlsx and rename the file as sampleAnswersS.xlsx. The time difference was converted from hours to number of days. The number of days were considered as speed of an answer (here 1 hour would be 0.042 days). Reputation of a user can be accessed once the user's Id was known from the Users table. At last, we divided the sampleAnswersS.xlsx into sampleTimesAA.xlsx and sample-TimesNA.xlsx. Where sampleTimesAA.xlsx was made using sampleAnswersAA.xlsx and contains SpeedM values of accepted answers and sampleTimesNA.xlsx was created using sample-AnswersNA.xlsx and contains SpeedM values of non-accepted answers.

We would utilize Kolmogorov–Smirnov test (KS – test) mentioned in [9] and [12] to examine the time difference between the question posted and answer received and the user reputation's effect on accepted answer. The KS - test calculates difference between two cumulative distribution functions of the two data sets. The KS – test is implemented by utilizing Pythons scipy.stats.ks_2samp module which has a function ks_2samp(x, y) with column x and column y as input parameters and KS and p value as returned value. We apply KS – test on two data sets where one contains accepted answer's user reputation (sampleProfilesAA.xlsx) and the other has user reputation of answers that were not accepted (sampleProfilesNA.xlsx). The acceptance of null hypothesis regards that acceptance of a solution is not dependent on user reputation. Similarly, we would use KS - test on accepted answer's speed (sampleTimesAA.xlsx) and non-accepted answer's speed (sampleTimesNA.xlsx). The results should be at a confidence level of 0.05 to be categorized as statistically significant this was calculated using the website Survey System https://www.surveysystem.com/sscalc.htm . The p-value from KS-test must less than 0.05 to consider results as valid and have a 95% confidence level in the observational difference.

## 5 FINDING AND DISCUSSION

### 5.1 RQ1

Out of 1000 samples in MathOverflow and Mathematics, 74.8% and 70.3% of questions had their accepted answer as the highest voted (maximum Score value) answers respectively. The accepted answer was not considered to have a maximum Score if all answers of a question had Score 0 including accepted answer. An answer with at least one vote and a highest Score was considered to be an answer with a maximum Score.

Approximately 75% questions in MathOverflow and 70% questions in Mathematics had their top voted answers as accepted answers. This could indicate a few different scenarios. The first scenario is that the number of votes on an answer massively effect the questioner's decision-making ability. The second scenario is that the accepted answer was selected by the questioner based on whether it was beneficial and useful to them or not. Therefore,

other users except the questioner that were looking for a solution to a similar question upvoted the accepted answer (chosen by the questioner) because they also found it useful for their problem. The last scenario is that the popularity of the posted question was very low due to which the questioner is the only one to upvote an answer that he/she accepted. Therefore, the answer with one vote is highest voted answer compared to the zeros of other answers. Each of these aspects give reasons for a large percentage of accepted answers were highest upvoted answers.

## 5.2 RQ2

Using the Kolmogorov-Smirnov test, for user reputation KS statistic value of 0.1975 and 0.1912 for MathOverflow and Mathematics respectively for user reputation. Furthermore, p value for user reputation was $9.016 * 10^{-13}$ and $2.156 * 10^{-13}$ MathOverflow and Mathematics respectively. As both p value mentioned above were less than 0.05 thus, we can say with 95% confidence level that the Reputation difference between accepted answers and non-accepted answers are statistically important and therefore, we can dismiss the null hypothesis. Nevertheless, our concentration is on Reputation greater than 10000 (which is high user reputation). After applying this filter, we get KS statistic value 0.0886 and 0.0831 along with p value as 0.3441 and 0.1107 for MathOverflow and Mathematics respectively. As latest p values were greater than 0.05, thus, we do not have sufficient evidence to reject null hypothesis. Hence, the acceptance of an answer is independent of high reputation of user.

For speed, the Kolmogorov-Smirnov test was utilized which gave KS statistic value of 0.1495 and 0.1245 for MathOverflow and Mathematics respectively. Additionally, p value for CreationDate difference (speed) between questions and answers were $7.501 * 10^{-13}$ and $5.362 * 10^{-9}$ for MathOverflow and Mathematics respectively. As both p values above are less than 0.05 therefore, we can reject the null hypothesis and can say with 95% confidence level that difference between speed of accepted answers and non-accepted answers are statistically significant. However, we only have to concentrate on fast replies, therefore, we only consider speed below 7 (days) (which is a week). Then, we got KS statistic value of 0.0434 and 0.0511 along with p value of 0.4412 and 0.7408 for MathOverflow and Mathematics respectively. Thus, both of the latest p values mentioned above are greater than 0.05 which provides us with enough proof to accept the null hypothesis. Hence, if the answer is posted within 7 days of question being posted then there is no significant relation between fast reply and acceptance of answer.

The results received above for high reputation (reputation greater than 10000) and high speed (speed less than 7) were not surprising as queries posted in MathOverflow and Mathematics are kept open for other users even after an answer is accepted. From the sampleAnswersS.xlsx of both MathOverflow and Mathematics we got a speed of over 3000 which is equivalent to 8 years approximately from the posting of the query. This probably skewed the results we received from first test (Reputation) and second test (Speed). According to the results from first (Reputation) and second (Speed) test after capping the values at 10000 and 7 respectively we conclude null hypothesis is valid in both tests.

# 6 THREATS TO VALIDITY

The goal of the research paper was to examine the potential effect of traits of an answer on acceptance of it. The answers posted by individuals generally represent their personality [3][11]. Mathematics and MathOverflow have a reputation score to somehow represent the reputation of the user [2]. However according to Yang et al. [15], reputation is generally dependent on various behaviours on Mathematics and MathOverflow. Expertise does not inevitably have anything to do with reputation rather activity of the user on the website is more related to reputation. This could be a threat to validity of the results we received in this research paper. In the “Finding and Discussion” heading, we noted that there might be another threat to validity which is the selection of type of question to be examined. Some posted questions had replies that only had one vote amongst three other answers. This leads to lack of questions that generate alluring findings and cause debates.

There are some internal threats to validity of the results. The creation MySQL database and use of SQL queries such as LOAD XML with correct syntax to load datasets in the Power BI for the first time are some possible confounding variables. Moreover, computation time is another confounding variable which led to time wastage due to iteration through 1.4 million questions in Mathematics [1] and 126 thousand questions in MathOverflow [1] to cure data . The data curation (that is SRS sampling) was needed to analyse relevant questions in both sites.

Some more threats to validity of the results are external ones. The results were generated using 1000 samples of Questions.xlsx in this research paper. Although, all samples were randomly and homogeneously selected from the Questions.xlsx, Answers.xlsx and Profiles.xlsx, however, the data can be distorted because data more than 95% was not contained in the 1000 samples of Questions.xlsx. If the 1000 samples are generated again then we might get different findings and thus this research cannot be generalized beyond the immediate study. Another external threat to validity is the claim to generalize the study might not be supported by the characteristics of accepted answer because all users are different. Also, the acceptance of answer by a particular questioner may vary from one question to another according to his/her agenda. A question with a simple syntax might receive an answer instantly, however, a complex question involving a particular algorithm might end being debated a lot. Hence, there are multiple parameters in play due to which the generalization of correlation between question and answer is difficult.

# 7 RELATED WORK

Yao Lu et al. [10] conducted a research using regression and qualitative analysis to study impact of incentive systems on Fast Answers in StackOverflow. The data was retrieved from Stack Exchange Data Dump that contains information of 8.1 million

users. The overall findings concluded that, late responses to posted question have a less probability to be accepted and voted. Therefore, the relation between fast replies and acceptance of an answer needs to be investigated to answer RQ2.

Consistent with Yao Lu et al. [10] relation between Fast Answers and gamification was highlighted in the study conducted by Shaowei Wang et al. [14]. The study used a logistic regression model on the Stack Exchange Data Dump to examine 46 factors for four websites of Stack Exchange which concluded that time required for accepting an answer was strongly affected by 46 factors under the answerer dimension. Hence, the rapid answering of questions is an important factor to verify RQ2.

In order to explore a relationship between the number of votes on a reply and the accepted answer a research was conducted by Neelamadhav Gantayat et al. [7]. Use of StackOverflow user post data has provided that 13.07% of questions with multiple answers had their accepted answer which were the highest voted answer. Thus, the relationship between number of votes and acceptance of answer is important factor to evaluate RQ1.

Yla R. Tausczik and James W. Pennebaker [13] analysed effect of online and offline user reputation on quality of a solution in MathOverflow. The study utilized linear and multi-level logistic regression on the MathOverflow Data Dump which suggested that quality of solution submission can be identified by the user's offline and online reputation. Therefore, user reputation is essential parameter for the verification of RQ2.

Keith Burghardt et al. [5], [6] conducted a study to identify the components that affect the possibility of an answer to be selected as a highest quality answer. The data from 250 websites of Stack Exchange was examined which lead to the conclusion that questioner accepts the answer based on heuristics to a greater extent as compared to voters. Moreover, the upvotes on an answer by the voters are affected if the answer is already accepted. Thus, voting of answer and acceptance of an answer are essential determinants to examine RQ1.

## 8 CONCLUSION

Mathematics is a big CQA platform that seek out mathematicians of all levels and professionals in related fields, while MathOverflow is only for professional mathematicians. Users have the ability to up vote or down vote an answer. Moreover, the questioner has an additional ability to accept an answer (which is an helpful answer). After performing the data analysis, 70 % and 75% of answers in Mathematics and MathOverflow respectively were found to have highest voted answers as accepted answers. Additionally, if the questioner found an answer helpful (accepted answer), it is probable that other users also found it beneficial and up voted the answer. It was found that user reputation of the answerer and the speed of an answer had no effect on the acceptance of an answer.

## 9 FUTURE WORK

Future research would involve datasets that will have more than 1000 sample data points. If in future work sufficient amount of computation power is provided, then the analysis of whole "Posts.xml" and "Users.xml" file can be performed. Another characteristic of accepted answer is the semantics of answers posted on Mathematics and MathOverflow can be explored. Some of these semantics are answer's length, spelling errors or MathJax (JavaScript display engine for mathematics) syntax error. To analyse this trait (semantics) of accepted answers coding analysis will be performed on the body of the answer posted.

## 10 APPENDIX

Following are the code snippets that were utilized to do data analysis for RQ1 and RQ2:

```
SET GLOBAL local_infile=1;

USE mathematics;
CREATE TABLE IF NOT EXISTS posts (
    IndexId INT NOT NULL AUTO_INCREMENT,
    Id INT,
    PostTypeId INT,
    AcceptedAnswerId INT,
    CreationDate TIMESTAMP,
    Score INT,
    ViewCount INT,
    Body TEXT,
    OwnerUserId INT,
    LastEditorUserId INT,
    LastEditorDisplayName TEXT,
    LastEditDate TIMESTAMP,
    LastActivityDate TIMESTAMP,
    Title TEXT,
    Tags TEXT,
    AnswerCount INT,
    CommentCount INT,
    FavoriteCount INT,
    ContentLicense TEXT,
    CommunityOwnedDate TIMESTAMP,
    ParentId INT,
    ClosedDate TIMESTAMP,
    OwnerDisplayName Text,
    PRIMARY KEY (IndexId)
);
LOAD XML LOCAL INFILE 'D:/UoA/Sem2/Research Methods in CS and SE/Posts.xml'
    INTO TABLE posts
    ROWS IDENTIFIED BY '<row>';
```

**Figure 1: Creating a "posts" table to fill values of "Posts.xml"**

```
CREATE TABLE p1 LIKE posts;
CREATE TABLE p2 LIKE posts;
CREATE TABLE p3 LIKE posts;
CREATE TABLE p4 LIKE posts;
CREATE TABLE p5 LIKE posts;
```

**Figure 2: Creating 5 tables in MySQL to split data of "posts" table**

```
INSERT INTO p1
SELECT *
FROM posts
WHERE IndexId >= 1
AND IndexId <= 808590;
```

**Figure 3: Inserting a subset of values from “posts” table to “p1” table**

```
INSERT INTO p2
SELECT *
FROM posts
WHERE IndexId >= 808591
AND IndexId <= 1617179;
```

**Figure 4: Inserting a subset of values from “posts” table to “p2” table**

```
INSERT INTO p3
SELECT *
FROM posts
WHERE IndexId >= 1617180
AND IndexId <= 2425769;
```

**Figure 5: Inserting a subset of values from “posts” table to “p3” table**

```
INSERT INTO p4
SELECT *
FROM posts
WHERE IndexId >= 2425770
AND IndexId <= 2830064;
```

**Figure 6: Inserting a subset of values from “posts” table to “p4” table**

```
INSERT INTO p5
SELECT *
FROM posts
WHERE IndexId >= 2830065
AND IndexId <= 3234358;
```

**Figure 7: Inserting a subset of values from “posts” table to “p5” table**

```
import pandas as pd
sampleQuestions1 = pd.read_excel ('sampleQ 1.xlsx')
sampleQuestions2 = pd.read_excel ('sampleQ 2.xlsx')
sampleQuestions3 = pd.read_excel ('sampleQ 3.xlsx')
sampleQuestions4 = pd.read_excel ('sampleQ 4.xlsx')
sampleQuestions5 = pd.read_excel ('sampleQ 5.xlsx')
ap1 = sampleQuestions1.append(sampleQuestions2, ignore_index = True)
ap2 = sampleQuestions3.append(sampleQuestions4, ignore_index = True)
ap3 = ap1.append(ap2, ignore_index = True)
sampleQuestions = ap3.append(sampleQuestions5, ignore_index = True)
sampleAnswers1 = pd.read_excel ('sampleA 1.xlsx')
sampleAnswers2 = pd.read_excel ('sampleA 2.xlsx')
sampleAnswers3 = pd.read_excel ('sampleA 3.xlsx')
sampleAnswers4 = pd.read_excel ('sampleA 4.xlsx')
sampleAnswers5 = pd.read_excel ('sampleA 5.xlsx')
aps1 = sampleAnswers1.append(sampleAnswers2, ignore_index = True)
aps2 = sampleAnswers3.append(sampleAnswers4, ignore_index = True)
aps3 = aps1.append(aps2, ignore_index = True)
sampleAnswers = aps3.append(sampleAnswers5, ignore_index = True)
n = sampleQuestions.shape[0]
n1 = 0
AnswerIdF = pd.DataFrame(columns = ['Id'])
for i in range(0, n):
    AnswerId = sampleAnswers['ParentId'].where(sampleAnswers['ParentId'] ==
sampleQuestions['Id'].iloc[i])
    AnswerId = AnswerId.dropna()
    AnswerIdU = sampleAnswers[['Score','ParentId']].iloc[AnswerId.index]
    if AnswerIdU.empty:
        AnswerIdU.loc[0,'Id'] = 0
        AnswerIdU.loc[0,'Score'] = 0
        n1 = n1+1
        AnswerIdF.loc[i, 'Id'] = 0
    else:
        AnswerIdA = AnswerIdU.idxmax()
        AnswerIdF.loc[i, 'Id'] = sampleAnswers['Id'].iloc[AnswerIdA['Score']]
IdFin = pd.DataFrame(columns = ['Id'])
j = 0
for i in range(0, n):
    if AnswerIdF['Id'].iloc[i] == sampleQuestions['AcceptedAnswerId'].iloc[i]:
        IdFin.loc[j, 'Id'] = AnswerIdF['Id'].iloc[i]
        j = j+1
n = n-n1
Per = IdFin.shape[0]/n
Per = Per*100
Per = round(Per,1)
print("\n")
print("The Highest Voted Answer that were Accepted are: ",Per)
print("\n")
```

**Figure 8: Python code to get percentage of highest upvoted answers that were accepted answers in Mathematics**

```
import pandas as pd
sampleQuestions = pd.read_excel ('sampleQuestions2.xlsx')
sampleAnswers = pd.read_excel ('sampleAnswers2.xlsx')
n = sampleQuestions.shape[0]
AnswerIdF = pd.DataFrame(columns = ['Id'])
for i in range(0, n):
    AnswerId = sampleAnswers['Attribute:ParentId'].where(sampleAnswers['Attribute:ParentId'] ==
sampleQuestions['Attribute:Id'].iloc[i])
    AnswerId = AnswerId.dropna()
    AnswerIdU = sampleAnswers[['Attribute:Score','Attribute:ParentId']].iloc[AnswerId.index]
    AnswerIdA = AnswerIdU.idxmax()
    AnswerIdF.loc[i, 'Id'] = sampleAnswers['Attribute:Id'].iloc[AnswerIdA['Attribute:Score']]
IdFin = pd.DataFrame(columns = ['Id'])
j = 0
for i in range(0, n):
    if AnswerIdF['Id'].iloc[i] == sampleQuestions['Attribute:AcceptedAnswerId'].iloc[i]:
        IdFin.loc[j, 'Id'] = AnswerIdF['Id'].iloc[i]
        j = j+1
Per = IdFin.shape[0]/n
print("\n")
print("The Highest Voted Answer that were Accepted are: ",Per*100)
print("\n")
```

**Figure 9: Python code to get percentage of highest upvoted answers that were accepted answers in MathOverflow**

```
import pandas as pd
sampleQuestions = pd.read_excel ('CreationDateData\sampleQuestions.xlsx')
sampleAnswers = pd.read_excel ('CreationDateData\sampleAnswers.xlsx')
import datetime as dt
n = sampleQuestions.shape[0]
m = sampleAnswers.shape[0]
sampleAnswers['Speed'] = sampleAnswers['CreationDate']
for i in range(0, n):
    for j in range(0, m):
        if sampleQuestions['Id'].iloc[i] == sampleAnswers['ParentId'].iloc[j] :
            sampleAnswers.loc[j, 'Speed'] = sampleAnswers['CreationDate'].iloc[j] -
sampleQuestions['CreationDate'].iloc[i]
for k in range(0, m):
    sampleAnswers.loc[k, 'Speed'] = sampleAnswers['Speed'].iloc[k].total_seconds()
sampleAnswers.to_excel("sampleAnswersS.xlsx")
```

**Figure 10: Python code to get sampleAnswersS.xlsx file for Mathematics**

```
import pandas as pd
sampleQuestions = pd.read_excel ('CreationDateData\sampleQuestions.xlsx')
sampleAnswers = pd.read_excel ('CreationDateData\sampleAnswers.xlsx')
import datetime as dt
n = sampleQuestions.shape[0]
m = sampleAnswers.shape[0]
sampleAnswers['Attribute:Speed'] = sampleAnswers['Attribute:CreationDate']
for i in range(0, n):
    for j in range(0, m):
        if sampleQuestions['Attribute:Id'].iloc[i] == sampleAnswers['Attribute:ParentId'].iloc[j] :
            sampleAnswers.loc[j, 'Attribute:Speed'] = sampleAnswers['Attribute:CreationDate'].iloc[j] -
sampleQuestions['Attribute:CreationDate'].iloc[i]
for k in range(0, m):
    sampleAnswers.loc[k, 'Attribute:Speed'] = sampleAnswers['Attribute:Speed'].iloc[k].total_seconds()
sampleAnswers.to_excel("sampleAnswersS.xlsx")
```

**Figure 11: Python code to get sampleAnswersS.xlsx file for MathOverflow**

```
import pandas as pd
sampleTimesAA = pd.read_excel ('sampleTimesAA.xlsx')
sampleTimesNA = pd.read_excel ('sampleTimesNA.xlsx')
import scipy.stats as ks
ks_t = ks.ks_2samp(sampleTimesAA['SpeedM'],sampleTimesNA['SpeedM'])
print(ks_t)
```

**Figure 12: Python code to get speed difference without any filter on data for MathOverflow and Mathematics**

```
import pandas as pd
sampleTimesAA = pd.read_excel ('sampleTimesAA.xlsx')
sampleTimesNA = pd.read_excel ('sampleTimesNA.xlsx')
import scipy.stats as ks
SpeedM1 = sampleTimesAA['SpeedM'].where((sampleTimesAA['SpeedM'] < 7) & (sampleTimesAA['SpeedM'] >
0.05))
SpeedM2 = sampleTimesNA['SpeedM'].where((sampleTimesNA['SpeedM'] < 7) & (sampleTimesNA['SpeedM'] >
0.05))
SpeedM1 = SpeedM1.dropna()
SpeedM2 = SpeedM2.dropna()
ks_t = ks.ks_2samp(SpeedM1,SpeedM2)
print(ks_t)
```

**Figure 13: Python code to get speed difference with filter on data for MathOverflow and Mathematics**

```
import pandas as pd
sampleProfilesAA = pd.read_excel ('sampleProfilesAA.xlsx')
sampleProfilesNA = pd.read_excel ('sampleProfilesNA.xlsx')
import scipy.stats as ks
ks_t = ks.ks_2samp(sampleProfilesAA['Attribute:Reputation'],sampleProfilesNA['Attribute:Reputation'])
print(ks_t)
```

**Figure 14: Python code to get reputation difference without any filter on data for MathOverflow and Mathematics**

```
import pandas as pd
sampleProfilesAA = pd.read_excel ('sampleProfilesAA.xlsx')
sampleProfilesNA = pd.read_excel ('sampleProfilesNA.xlsx')
import scipy.stats as ks
Rep1 = sampleProfilesAA['Attribute:Reputation'].where(sampleProfilesAA['Attribute:Reputation'] > 10000)
Rep2 = sampleProfilesNA['Attribute:Reputation'].where(sampleProfilesNA['Attribute:Reputation'] > 10000)
Rep1 = Rep1.dropna()
Rep2 = Rep2.dropna()
ks_t = ks.ks_2samp(Rep1,Rep2)
print(ks_t)
```

**Figure 15: Python code to get reputation difference with filter on data for MathOverflow and Mathematics**

# REFERENCES


[1] Jeff Atwood and Joel Spolsky. 2009. *All Sites - Stack Exchange*. Retrieved May 28, 2021 from https://stackexchange.com/sites#science-traffic

[2] Jeff Atwood and Joel Spolsky. 2009. *FAQ - Area 51 - Stack Exchange*. Retrieved May 28, 2021 from https://area51.stackexchange.com/faq

[3] Blerina Bazelli, Abram Hindle, and Eleni Stroulia. 2013. On the Personality Traits of StackOverflow Users. In *2013 IEEE International Conference on Software Maintenance*. 460–463. https://doi.org/10.1109/ICSM.2013.72

[4] Grégoire Burel, Yulan He, and Harith Alani. 2012. Automatic identification of best answers in online enquiry communities. In *Extended Semantic Web Conference*. Springer, 514–529.

[5] Keith Burghardt, Emanuel Alsina, Michelle Girvan, William Rand, and Kristina Lerman. 2016. The myopia of crowds: A study of collective evaluation on stack exchange. *Robert H. Smith School Research Paper No. RHS* 2736568(2016).

[6] Keith Burghardt, Emanuel F Alsina, Michelle Girvan, William Rand, and Kristina Lerman. 2017. The myopia of crowds: Cognitive load and collective evaluation of answers on Stack Exchange. *PloS one* 12, 3 (2017), e0173610.

[7] Neelamadhav Gantayat, Pankaj Dhoolia, Rohan Padhye, Senthil Mani, and Vibha Singhal Sinha. 2015. The synergy between voting and acceptance of answers on stackoverflow-or the lack thereof. In *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*. IEEE, 406–409.

[8] Anton Geraschenko and Scott Morrison. 2009. *MathOverflow*. Retrieved May 28, 2021 from https://sbseminar.wordpress.com/2009/10/14/math-overflow/

[9] Andrey Kolmogorov. 1933. Sulla determinazione empirica di una lgge di distribuzione. *Inst. Ital. Attuari, Giorn.* 4 (1933), 83–91.

[10] Yao Lu, Xinjun Mao, Minghui Zhou, Yang Zhang, Tao Wang, and Zude Li. 2020. Haste Makes Waste: An Empirical Study of Fast Answers in Stack Overflow. In *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 23–34.

[11] Ayushi Rastogi and Nachiappan Nagappan. 2016. On the Personality Traits of GitHub Contributors. In *2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE)*. 77–86. https://doi.org/10.1109/ISSRE.2016.43

[12] Nickolay Smirnov. 1948. Table for estimating the goodness of fit of empirical distributions. *The annals of mathematical statistics* 19, 2 (1948), 279–281.

[13] Yla R Tausczik and James W Pennebaker. 2011. Predicting the perceived quality of online mathematics contributions from users' reputations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1885–1888.

[14] Shaowei Wang, Tse-Hsun Chen, and Ahmed E Hassan. 2018. Understanding the factors for fast answers in technical Q&A websites. *Empirical Software Engineering* 23, 3 (2018), 1552–1593.

[15] Jie Yang, Ke Tao, Alessandro Bozzon, and Geert-Jan Houben. 2014. Sparrows and owls: Characterisation of expert behaviour in stackoverflow. In *International conference on user modeling, adaptation, and personalization*. Springer, 266–277.