CS108 Bash Grader Testcases and Explanation

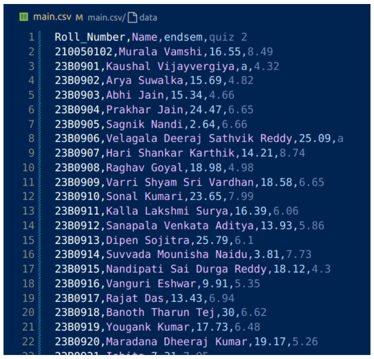
Aditya Neeraje : 23B0940

April 28, 2024

1 Initialize

Running the following command initializes 4 csv files with random data and combines some of this data into main.csv.

```
python3 initialize.py
```



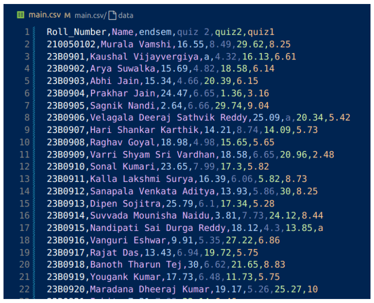
```
Roll_Number,Name,endsem,quiz 2
210050102,Murala Vamshi,16.55,8.49
23B0901,Kaushal Vijayvergiya,a,4.32
23B0902,Arya Suwalka,15.69,4.82
23B0903,Abhi Jain,15.34,4.66
23B0904,Prakhar Jain,24.47,6.65
23B0905,Sagnik Nandi,2.64,6.66
23B0906,Velagala Deeraj Sathvik Reddy,25.09,a
23B0907,Hari Shankar Karthik,14.21,8.74
23B0908,Raghav Goyal,18.98,4.98
23B0909,Varri Shyam Sri Vardhan,18.58,6.65
23B0910,Sonal Kumari,23.65,7.99
23B0911,Kalla Lakshmi Surya,16.39,6.06
23B0912,Sanapala Venkata Aditya,13.93,5.86
23B0913,Dipen Sojitra,25.79,6.1
23B0914,Suvvada Mounisha Naidu,3.81,7.73
23B0915,Nandipati Sai Durga Reddy,18.12,4.3
23B0916,Vanguri Eshwar,9.91,5.35
23B0917,Rajat Das,13.43,6.94
23B0918,Banoth Tharun Tej,30,6.62
23B0919,Yougank Kumar,17.73,6.48
23B0920,Maradana Dheeraj Kumar,19.17,5.26
```

Figure 1: main.csv after Initializing

2 Combine

Running the following command combines the data from 4 csv files into main.csv.

```
bash submission.sh combine
```



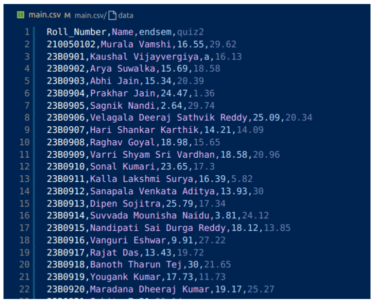
```
Roll_Number,Name,endsem,quiz 2,quiz2,quiz1
210050102,Murala Vamshi,16.55,8.49,29.62,8.25
23B0901,Kaushal Vijayvergiya,a,4.32,16.13,6.61
23B0902,Arya Suwalka,15.69,4.82,18.58,6.14
23B0903,Abhi Jain,15.34,4.66,20.39,6.15
23B0904,Prakhar Jain,24.47,6.65,1.36,3.16
23B0905,Sagnik Nandi,2.64,6.66,29.74,9.04
23B0906,Velagala Deeraj Sathvik Reddy,25.09,a,20.34,5.42
23B0907,Hari Shankar Karthik,14.21,8.74,14.09,5.73
23B0908,Raghav Goyal,18.98,4.98,15.65,5.65
23B0909,Varri Shyam Sri Vardhan,18.58,6.65,20.96,2.48
23B0910,Sonal Kumari,23.65,7.99,17.3,5.82
23B0911,Kalla Lakshmi Surya,16.39,6.06,5.82,8.73
23B0912,Sanapala Venkata Aditya,13.93,5.86,30,8.25
23B0913,Dipen Sojitra,25.79,6.1,17.34,5.28
23B0914,Suvvada Mounisha Naidu,3.81,7.73,24.12,8.44
23B0915,Nandipati Sai Durga Reddy,18.12,4.3,13.85,a
23B0916,Vanguri Eshwar,9.91,5.35,27.22,6.86
23B0917,Rajat Das,13.43,6.94,19.72,5.75
23B0918,Banoth Tharun Tej,30,6.62,21.65,8.83
23B0919,Yougank Kumar,17.73,6.48,11.73,5.75
23B0920,Maradana Dheeraj Kumar,19.17,5.26,25.27,10
```

Figure 2: main.csv after combining all 4

You can specify files to drop by passing them as arguments to the script.

```
bash submission.sh combine --drop quiz1.csv "quiz 2.csv"
bash submission.sh combine -d quiz1.csv "quiz 2.csv"
```

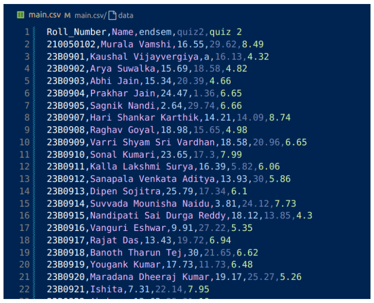
You can selectively combine back certain quizzes alone by passing them as arguments before any other flag.



```
Roll_Number,Name,endsem,quiz2
210050102,Murala Vamshi,16.55,29.62
23B0901,Kaushal Vijayvergiya,a,16.13
23B0902,Arya Suwalka,15.69,18.58
23B0903,Abhi Jain,15.34,20.39
23B0904,Prakhar Jain,24.47,1.36
23B0905,Sagnik Nandi,2.64,29.74
23B0906,Velagala Deeraj Sathvik Reddy,25.09,20.34
23B0907,Hari Shankar Karthik,14.21,14.09
23B0908,Raghav Goyal,18.98,15.65
23B0909,Varri Shyam Sri Vardhan,18.58,20.96
23B0910,Sonal Kumari,23.65,17.3
23B0911,Kalla Lakshmi Surya,16.39,5.82
23B0912,Sanapala Venkata Aditya,13.93,30
23B0913,Dipen Sojitra,25.79,17.34
23B0914,Suvvada Mounisha Naidu,3.81,24.12
23B0915,Nandipati Sai Durga Reddy,18.12,13.85
23B0916,Vanguri Eshwar,9.91,27.22
23B0917,Rajat Das,13.43,19.72
23B0918,Banoth Tharun Tej,30,21.65
23B0919,Yougank Kumar,17.73,11.73
23B0920,Maradana Dheeraj Kumar,19.17,25.27
```

Figure 3: main.csv after dropping quiz1.csv and quiz 2.csv

```
bash submission.sh combine quiz1.csv
```



```
Roll_Number,Name,endsem,quiz2,quiz 2
210050102,Murala Vamshi,16.55,29.62,8.49
23B0901,Kaushal Vijayvergiya,a,16.13,4.32
23B0902,Arya Suwalka,15.69,18.58,4.82
23B0903,Abhi Jain,15.34,20.39,4.66
23B0904,Prakhar Jain,24.47,1.36,6.65
23B0905,Sagnik Nandi,2.64,29.74,6.66
23B0907,Hari Shankar Karthik,14.21,14.09,8.74
23B0908,Raghav Goyal,18.98,15.65,4.98
23B0909,Varri Shyam Sri Vardhan,18.58,20.96,6.65
23B0910,Sonal Kumari,23.65,17.3,7.99
23B0911,Kalla Lakshmi Surya,16.39,5.82,6.06
23B0912,Sanapala Venkata Aditya,13.93,30,5.86
23B0913,Dipen Sojitra,25.79,17.34,6.1
23B0914,Suvvada Mounisha Naidu,3.81,24.12,7.73
23B0915,Nandipati Sai Durga Reddy,18.12,13.85,4.3
23B0916,Vanguri Eshwar,9.91,27.22,5.35
23B0917,Rajat Das,13.43,19.72,6.94
23B0918,Banoth Tharun Tej,30,21.65,6.62
23B0919,Yougank Kumar,17.73,11.73,6.48
23B0920,Maradana Dheeraj Kumar,19.17,25.27,5.26
23B0921,Ishita,7.31,22.14,7.95
```

Figure 4: main.csv after combining quiz 2.csv

Generally, combine does not recheck or recombine quizzes which already have a column assigned to them in main.csv, and whose first 3 lines of data are valid marks. Use the –force flag to combine them.

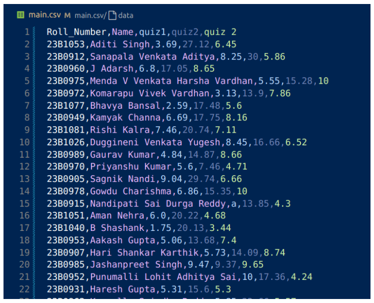
```
bash submission.sh combine quiz1.csv quiz2.csv "quiz 2.csv" --force
```

Note here that endsem.csv is actually lost from main, since both the force_flag and the only_flag are true and endsem.csv is not amongst the list of quizzes to only add to main.

combine also works as intended if main.csv is not present in the `WORKING_DIRECTORY`, creating main.csv. All flags work similarly as to the normal case. The user might notice non-fatal errors being registered due to head trying to read the non-existent main.csv file. This is expected behavior. Similarly, combine works as intended even if main.csv is empty (such as by running `echo "" > main.csv`).

3 Rescale

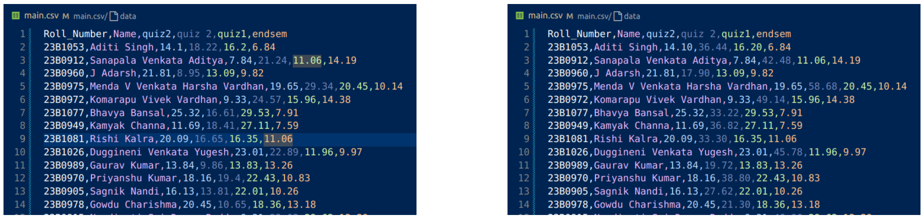
Rescale can accept arguments as a list of weights, or as a custom series of quiz_names and weights. The first way can be achieved through the `-w`/`--weights-only` flag, or by passing no flag.



```
Roll_Number,Name,quiz1,quiz2,quiz 2
23B1053,Aditi Singh,3.69,27.12,6.45
23B0912,Sanapala Venkata Aditya,8.25,30,5.86
23B0960,J Adarsh,6.8,17.05,8.65
23B0975,Menda V Venkata Harsha Vardhan,5.55,15.28,10
23B0972,Komarapu Vivek Vardhan,3.13,13.9,7.86
23B1077,Bhavya Bansal,2.59,17.48,5.6
23B0949,Kamyak Channa,6.69,17.75,8.16
23B1081,Rishi Kalra,7.46,20.74,7.11
23B1026,Duggineni Venkata Yugesh,8.45,16.66,6.52
23B0989,Gaurav Kumar,4.84,14.87,8.66
23B0970,Priyanshu Kumar,5.6,7.46,4.71
23B0905,Sagnik Nandi,9.04,29.74,6.66
23B0978,Gowdu Charishma,6.86,15.35,10
23B0915,Nandipati Sai Durga Reddy,a,13.85,4.3
23B1051,Aman Nehra,6.0,20.22,4.68
23B1040,B Shashank,1.75,20.13,3.44
23B0953,Aakash Gupta,5.06,13.68,7.4
23B0907,Hari Shankar Karthik,5.73,14.09,8.74
23B0985,Jashanpreet Singh,9.47,9.37,9.65
23B0952,Punumalli Lohit Adhitya Sai,10,17.36,4.24
23B0931,Haresh Gupta,5.31,15.6,5.3
```

Figure 5: main.csv after combining all but endsem with force

```
bash submission.sh rescale -w 1 2 1 1
```



```
Roll_Number,Name,quiz2,quiz 2,quiz1,endsem
23B1053,Aditi Singh,14.1,18.22,16.2,6.84
23B0912,Sanapala Venkata Aditya,7.84,21.24,11.06,14.19
23B0960,J Adarsh,21.81,8.95,13.09,9.82
23B0975,Menda V Venkata Harsha Vardhan,19.65,29.34,20.45,10.14
23B0972,Komarapu Vivek Vardhan,9.33,24.57,15.96,14.38
23B1077,Bhavya Bansal,25.32,16.61,29.53,7.91
23B0949,Kamyak Channa,11.69,18.41,27.11,7.59
23B1081,Rishi Kalra,20.09,16.65,16.35,11.06
23B1026,Duggineni Venkata Yugesh,23.01,22.89,11.96,9.97
23B0989,Gaurav Kumar,13.84,9.86,13.83,13.26
23B0970,Priyanshu Kumar,18.16,19.4,22.43,10.83
23B0905,Sagnik Nandi,16.13,13.81,22.01,10.26
23B0978,Gowdu Charishma,20.45,10.65,18.36,13.18
```

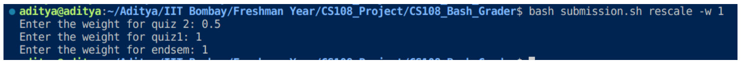
(a) Before Rescale

```
Roll_Number,Name,quiz2,quiz 2,quiz1,endsem
23B1053,Aditi Singh,14.10,36.44,16.20,6.84
23B0912,Sanapala Venkata Aditya,7.84,42.48,11.06,14.19
23B0960,J Adarsh,21.81,17.90,13.09,9.82
23B0975,Menda V Venkata Harsha Vardhan,19.65,58.68,20.45,10.14
23B0972,Komarapu Vivek Vardhan,9.33,49.14,15.96,14.38
23B1077,Bhavya Bansal,25.32,33.22,29.53,7.91
23B0949,Kamyak Channa,11.69,36.82,27.11,7.59
23B1081,Rishi Kalra,20.09,33.30,16.35,11.06
23B1026,Duggineni Venkata Yugesh,23.01,45.78,11.96,9.97
23B0989,Gaurav Kumar,13.84,19.72,13.83,13.26
23B0970,Priyanshu Kumar,18.16,38.80,22.43,10.83
23B0905,Sagnik Nandi,16.13,27.62,22.01,10.26
23B0978,Gowdu Charishma,20.45,21.30,18.36,13.18
```

(b) After Rescale

Figure 6: Rescale with weights 1 2 1 1

If you do not pass as many weights as there are quizzes, the script will prompt for more weights in order of the quizzes' appearance in main.csv.

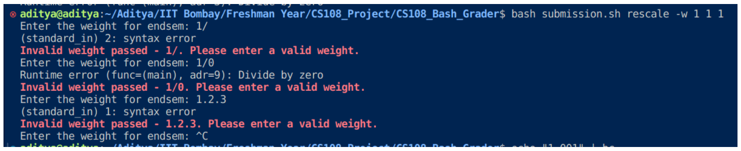


```
● aditya@aditya:~/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader$ bash submission.sh rescale -w 1
  Enter the weight for quiz 2: 0.5
  Enter the weight for quiz1: 1
  Enter the weight for endsem: 1
```

Figure 7: Prompting user for weights

It is worth noting that if the user enters invalid entries, the prompts will continue until a valid weight is given.
As is the default with the `bc` command on bash, non-numeric values such as "a" or "hello" get translated to 0, whereas invalid entries include entries such as "1.2.3" or "1/0" or "1/", etc.
The fact that rescale runs bc in the background on the weights inputted means that the weights can be in the form of expressions such as $2\hat{4}$, 1/3, etc. To pass longer expressions with spaces (such as $1 + 1 + 1$), either enclose the expressions in brackets or pass them using -c flag, which uses the `read` command instead of considering each word a new argument.

The `-c`/`--custom` flag allows you to pass a custom series of quiz names and weights. The script will assume the weights of the quizzes not mentioned in the



```
aditya@aditya:~/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader$ bash submission.sh rescale -w 1 1 1
Enter the weight for endsem: 1/
(standard_in) 2: syntax error
Invalid weight passed - 1/. Please enter a valid weight.
Enter the weight for endsem: 1/0
Runtime error (func=(main), adr=9): Divide by zero
Invalid weight passed - 1/0. Please enter a valid weight.
Enter the weight for endsem: 1.2.3
(standard_in) 1: syntax error
Invalid weight passed - 1.2.3. Please enter a valid weight.
Enter the weight for endsem: ^C
```

Figure 8: The script prompts the user until correct data is entered.

custom series are 1.

Quiz names are to be passed to the prompt as basenames only, without .csv extensions. If the same quiz name is passed twice, only the second corresponding weight is considered.

Ctrl+D is used to signal the end of the input.

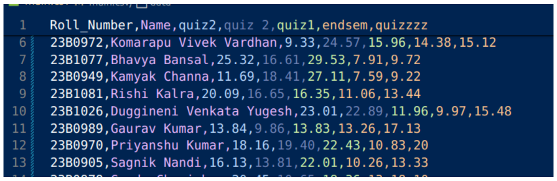
4 Upload

Upload is fairly simple. It checks for the existence of the given file, checks for validity of the data in the given file (i.e the header format and that all quiz marks must be numeric or ”a”). It prompts for user permission if the upload would overwrite a file already in the `WORKING_DIRECTORY`, unless the `--force` flag is passed.

By default, Upload calls Combine after uploading, as this is the expected usage of my script.

For example, if ../quizzz.csv is the file to be uploaded, the following command will upload the file and combine it with main.csv.

```
bash submission.sh upload ../quizzz.csv
```



```
Roll_Number,Name,quiz2,quiz 2,quiz1,endsem,quizzzz
23B0972,Komarapu Vivek Vardhan,9.33,24.57,15.96,14.38,15.12
23B1077,Bhavya Bansal,25.32,16.61,29.53,7.91,9.72
23B0949,Kamyak Channa,11.69,18.41,27.11,7.59,9.22
23B1081,Rishi Kalra,20.09,16.65,16.35,11.06,13.44
23B1026,Duggineni Venkata Yugesh,23.01,22.89,11.96,9.97,15.48
23B0989,Gaurav Kumar,13.84,9.86,13.83,13.26,17.13
23B0970,Priyanshu Kumar,18.16,19.40,22.43,10.83,20
23B0905,Sagnik Nandi,16.13,13.81,22.01,10.26,13.33
```

Figure 9: main.csv after uploading quizzz.csv

To remove an uploaded quiz, running

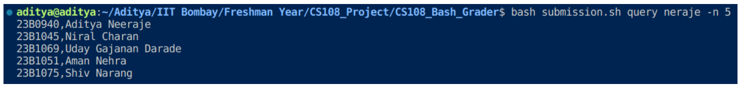
```
bash submission.sh combine --drop quizzz.csv
```

Of course, this does not delete the copy of quizzzz.csv in the `WORKING_DIRECTORY`, which will have to be done manually using rm.

5 Query

Finds the closest match either amongst Roll Numbers or the Names present in main.csv to the given query.
If the query is a substring of a name or roll number, it is considered a match, and all strings of which it is a substring are returned.
If it is not a substring of any name, the closest n matches are calculated using the Levenshtein distance between each word of the query and any word of the potential match.
For instance ”Aditya Neeraje” and ”Neeraje Aditya” have a Levenshtein distance of 0, since ”Aditya” and ”Aditya” are the same, and ”Neeraje” and ”Neeraje” are the same.
Using the `-n/--number` flag, you can specify the number of matches to return.
Using the `-u/--uniq` flag, you can specify that only the roll number of the best match is to be returned, which other functions such as percentile and update make use of to get the closest roll number to the given query.

```
bash submission.sh query Neraje -n 5
```



```
aditya@aditya:~/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader$ bash submission.sh query neraje -n 5
23B0940,Aditya Neeraje
23B1045,Niral Charan
23B1069,Uday Gajanan Darade
23B1051,Aman Nehra
23B1075,Shiv Narang
```

Figure 10: Querying for ”Neraje” instead of Neeraje with 5 matches

An arbitrary number of queries can be passed at once to query, and the script will return the closest match to each query.

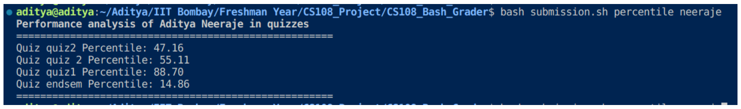
6 Percentile

Uses awk to calculate the percentile of a given student in each quiz. The spell-checking algorithm is run in the background.
You can use the -r flag to specify to what number of decimal places to round the percentile.

```
bash submission.sh percentile 23B0940 "Neeraje"
        "Aditya Neeraje" "Neeraje Aditya"
```

7 Analyze

Uses percentile to get the percentile of the student in every quiz, then finds his average percentile. If his percentile in any quiz is somewhat or significantly lower than his average percentile. Analyze reports this. If the student has been



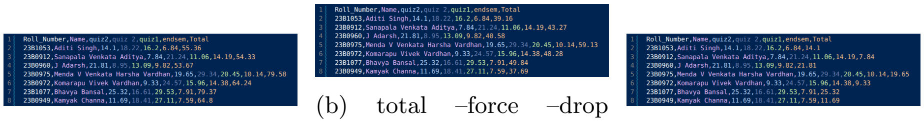
```
aditya@aditya:~/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader$ bash submission.sh percentile neeraje
Performance analysis of Aditya Neeraje in quizzes
====================================================
Quiz quiz2 Percentile: 47.16
Quiz quiz 2 Percentile: 55.11
Quiz quiz1 Percentile: 88.70
Quiz endsem Percentile: 14.86
====================================================
```

Figure 11: Percentile of Neeraje

consistent, average reports this.
A percentile drop of between 10% and 20% below average is taken to be somewhat underwhelming, and a percentile drop of greater than 20% is taken to be significant underperformance.

8 Total

Adds a total column to main.csv, with the columns to sum up determined by the user using the `-d`/`--drop` flag and the only flag, just like in combine.
Assuming main.csv initially has all 4 quizzes, the following commands will have the following results: Note that without the force flag, if a total column is present in main.csv and it has valid numeric or "a" marks, the script will not evaluate the total column again.



```
Roll_Number,Name,quiz2,quiz 2,quiz1,endsem,Total
23B1053,Aditi Singh,14.1,18.22,16.2,6.84,55.36
23B0912,Sanapala Venkata Aditya,7.84,21.24,11.06,14.19,54.33
23B0960,J Adarsh,21.81,8.95,13.09,9.82,53.67
23B0975,Menda V Venkata Harsha Vardhan,19.65,29.34,20.45,10.14,79.58
23B0972,Komarapu Vivek Vardhan,9.33,24.57,15.96,14.38,64.24
23B1077,Bhavya Bansal,25.32,16.61,29.53,7.91,79.37
23B0949,Kamyak Channa,11.69,18.41,27.11,7.59,64.8
```

(a) total –force

```
Roll_Number,Name,quiz2,quiz 2,quiz1,endsem,Total
23B1053,Aditi Singh,14.1,18.22,16.2,6.84,39.16
23B0912,Sanapala Venkata Aditya,7.84,21.24,11.06,14.19,43.27
23B0960,J Adarsh,21.81,8.95,13.09,9.82,40.58
23B0975,Menda V Venkata Harsha Vardhan,19.65,29.34,20.45,10.14,59.13
23B0972,Komarapu Vivek Vardhan,9.33,24.57,15.96,14.38,48.28
23B1077,Bhavya Bansal,25.32,16.61,29.53,7.91,49.84
23B0949,Kamyak Channa,11.69,18.41,27.11,7.59,37.69
```

(b) total –force –drop quiz1.csv

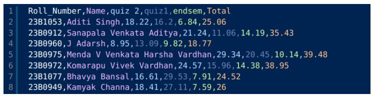
```
Roll_Number,Name,quiz2,quiz 2,quiz1,endsem,Total
23B1053,Aditi Singh,14.1,18.22,16.2,6.84,14.1
23B0912,Sanapala Venkata Aditya,7.84,21.24,11.06,14.19,7.84
23B0960,J Adarsh,21.81,8.95,13.09,9.82,21.81
23B0975,Menda V Venkata Harsha Vardhan,19.65,29.34,20.45,10.14,19.65
23B0972,Komarapu Vivek Vardhan,9.33,24.57,15.96,14.38,9.33
23B1077,Bhavya Bansal,25.32,16.61,29.53,7.91,25.32
23B0949,Kamyak Channa,11.69,18.41,27.11,7.59,11.69
```

(c) total quiz2.csv –force

Figure 12: main.csv after adding total

Note further that all commands, such as combine, upload, rescale and update keep track of which columns of main.csv were dropped earlier from total and drop them again while running total after the command is processed. For instance, running:

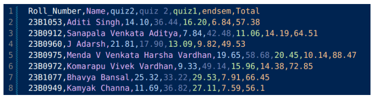
```
bash submission.sh total --force --drop quiz1.csv
bash submission.sh combine --drop quiz2.csv
```



```
Roll_Number,Name,quiz 2,quiz1,endsem,Total
23B1053,Aditi Singh,18.22,16.2,6.84,25.06
23B0912,Sanapala Venkata Aditya,21.24,11.06,14.19,35.43
23B0960,J Adarsh,8.95,13.09,9.82,18.77
23B0975,Menda V Venkata Harsha Vardhan,29.34,20.45,10.14,39.48
23B0972,Komarapu Vivek Vardhan,24.57,15.96,14.38,38.95
23B1077,Bhavya Bansal,16.61,29.53,7.91,24.52
23B0949,Kamyak Channa,18.41,27.11,7.59,26
```

Figure 13: Even after combine, the Total column excludes quiz1.csv

```
bash submission.sh total --force --drop quiz1.csv
bash submission.sh rescale -w 1 2 1 1
```

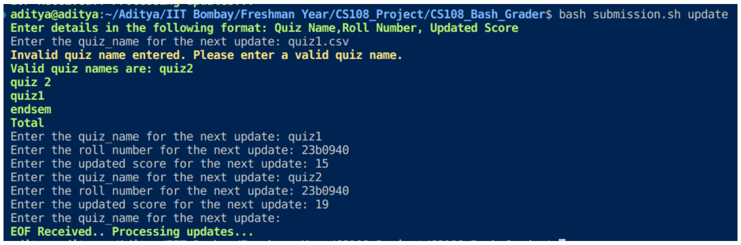


```
Roll_Number,Name,quiz2,quiz 2,quiz1,endsem,Total
23B1053,Aditi Singh,14.10,36.44,16.20,6.84,57.38
23B0912,Sanapala Venkata Aditya,7.84,42.48,11.06,14.19,64.51
23B0960,J Adarsh,21.81,17.90,13.09,9.82,49.53
23B0975,Menda V Venkata Harsha Vardhan,19.65,58.68,20.45,10.14,88.47
23B0972,Komarapu Vivek Vardhan,9.33,49.14,15.96,14.38,72.85
23B1077,Bhavya Bansal,25.32,33.22,29.53,7.91,66.45
23B0949,Kamyak Channa,11.69,36.82,27.11,7.59,56.1
```

Figure 14: Even after rescale, the Total column excludes quiz1.csv, but accounts for the doubling of quiz 2 marks

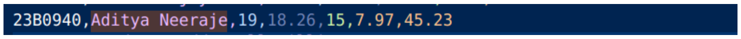
9 Update

Accepts the quiz name, a query which is then searched against the list of roll numbers and student names, and the updated marks. Total is also updated keeping in mind quizzes which were previously dropped.



```
aditya@aditya:~/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader$ bash submission.sh update
Enter details in the following format: Quiz Name,Roll Number, Updated Score
Enter the quiz_name for the next update: quiz1.csv
Invalid quiz name entered. Please enter a valid quiz name.
Valid quiz names are: quiz2
quiz 2
quiz1
endsem
Total
Enter the quiz_name for the next update: quiz1
Enter the roll number for the next update: 23b0940
Enter the updated score for the next update: 15
Enter the quiz_name for the next update: quiz2
Enter the roll number for the next update: 23b0940
Enter the updated score for the next update: 19
Enter the quiz_name for the next update:
EOF Received.. Processing updates...
```

Figure 15: Updating the marks of Neeraje



```
23B0940,Aditya Neeraje,19,18.26,15,7.97,45.23
```

Figure 16: main.csv after updating Neeraje's marks

Invalid details are handled as shown below:

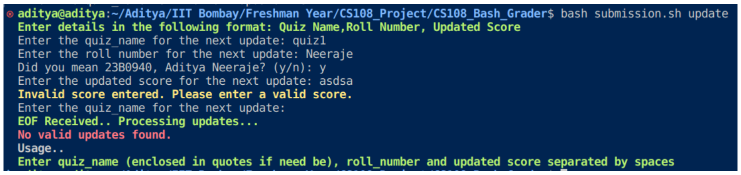
Note that update after rescale is a case I have not handled, as only the field corresponding to the quiz name and student in main.csv is updated, hence the scaling factor is not automatically applied. For now, it is recommended to simply run rescale at the end, after all cribs.

10 git_init

Initializes a git repository in the folder passed as an argument. A symlink called .my_git is created that points to the given directory.
I have implemented a for loop that allows for the directory to be created recursively even if some of the ancestors of the final directory mentioned do not exist. For example:

```
bash submission.sh git_init ../Aditya/Neeraje/Hello/World
```

generates the entire sequence of directories: Aditya, Neeraje, Hello and World even if Aditya does not exist in the ../ directory.



```
⊗ aditya@aditya:~/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader$ bash submission.sh update
  Enter details in the following format: Quiz Name,Roll Number, Updated Score
  Enter the quiz_name for the next update: quiz1
  Enter the roll number for the next update: Neeraje
  Did you mean 23B0940, Aditya Neeraje? (y/n): y
  Enter the updated score for the next update: asdsa
  Invalid score entered. Please enter a valid score.
  Enter the quiz_name for the next update:
  EOF Received.. Processing updates...
  No valid updates found.
  Usage..
  Enter quiz_name (enclosed in quotes if need be), roll_number and updated score separated by spaces
```

Figure 17: Invalid Update

For now, let us run the command:

```
bash submission.sh git_init Neeraje
```

This creates a directory Neeraje in `WORKING_DIRECTORY` and creates a symlink to it.

After at least one commit is made in the repository, calling git_init again will prompt before shifting repositories. The .my_git symlink now points to the new repository, which obtains all the commit history of the old repository.
Due to a logical error in having reversed the order of the following two lines:

```
grep -Ev "^$" "$WORKING_DIRECTORY/.my_git/.git_log"
    > "$WORKING_DIRECTORY/$TEMPORARY_FILE" 2>/dev/null
ln -s "$final_directory" "$WORKING_DIRECTORY/.my_git"
```

in the submission, the data stored in the .git_log file is not copied while reinitializing the git repository. That bug has since been corrected and pushed to my GitHub repository[1].
Also, due to another logical error, if the git repository had been initialized but the .git_log file was empty due to a lack of commits, running git_init again would not shift the git repository. This was because the deletion of the symlink occurs inside an if statement that requires the non-emptiness of the .git_log file to be entered. This error has been resolved as well.

11 git_commit

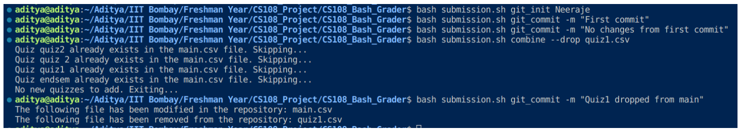
```
bash submission.sh git_commit -m "First commit - All Files"
```

Due to a logical error, git_commit does not copy and store main.csv. That has been modified by the use of the following lines in the latest version:

```
selected_quizzes+=("$WORKING_DIRECTORY/main.csv")
for quiz in "${quizzes[@]}"; do
    if [[ -f "$quiz" ]]; then
        selected_quizzes+=("$quiz")
    fi
done
```

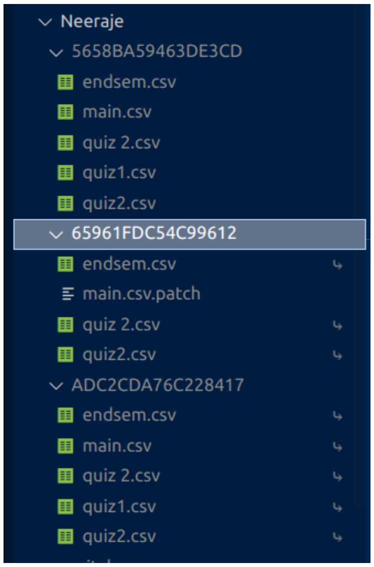
This ensures that main.csv is always included in the list of files to be committed.

As you can see, the first commit stores all the files as csv files. The second commit, which has no changes relative to the first, stores all files as symlinks. The third commit, where quiz1.csv is dropped, stores all files except main.csv as symlinks. main.csv is stored as a patch file.



```
aditya@aditya:~/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader$ bash submission.sh git_init Neeraje
aditya@aditya:~/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader$ bash submission.sh git_commit -m "First commit"
aditya@aditya:~/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader$ bash submission.sh git_commit -m "No changes from first commit"
aditya@aditya:~/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader$ bash submission.sh combine --drop quiz1.csv
Quiz quiz2 already exists in the main.csv file. Skipping...
Quiz quiz 2 already exists in the main.csv file. Skipping...
Quiz quiz1 already exists in the main.csv file. Skipping...
Quiz endsem already exists in the main.csv file. Skipping...
No new quizzes to add. Exiting...
aditya@aditya:~/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader$ bash submission.sh git_commit -m "Quiz1 dropped from main"
The following file has been modified in the repository: main.csv
The following file has been removed from the repository: quiz1.csv
```

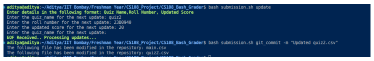
Figure 18: Some commands to generate commits



```
Neeraje
  5658BA59463DE3CD
    endsem.csv
    main.csv
    quiz 2.csv
    quiz1.csv
    quiz2.csv
  65961FDC54C99612
    endsem.csv
    main.csv.patch
    quiz 2.csv
    quiz2.csv
  ADC2CDA76C228417
    endsem.csv
    main.csv
    quiz 2.csv
    quiz1.csv
    quiz2.csv
```

Figure 19: How the files are stored

After this update, quiz2.csv is also stored as a patch file.

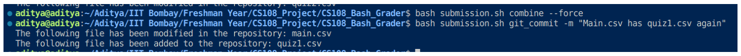


```
aditya@aditya:~/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader$ bash submission.sh update
Enter details in the following format: Quiz Name,Roll Number, Updated Score
Enter the quiz_name for the next update: quiz2
Enter the roll number for the next update: 23B0940
Enter the updated score for the next update: 20
Enter the quiz_name for the next update:
EOF Received.. Processing updates...
aditya@aditya:~/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader$ bash submission.sh git_commit -m "Updated quiz2.csv"
The following file has been modified in the repository: main.csv
The following file has been modified in the repository: quiz2.csv
```

Figure 20: Git commit after an update, shows that main.csv and quiz2.csv have been modified

git_commit –amend with an optional message allows you to amend the last commit. The last commit data is still stored, but it no longer shows up in git_log.

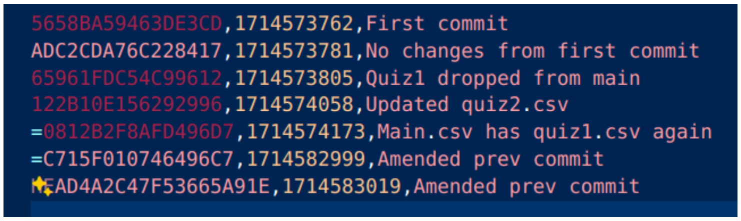
```
bash submission.sh git_commit --amend -m "Amended prev commit"
```



```
aditya@aditya:~/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader$ bash submission.sh combine --force
aditya@aditya:~/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader$ bash submission.sh git_commit -m "Main.csv has quiz1.csv again"
The following file has been modified in the repository: main.csv
The following file has been added to the repository: quiz1.csv
```

Figure 21: Git commit after force combining main.csv, shows that quiz1.csv has been added again. quiz1.csv is still a symlink in the remote.

```
bash submission.sh git_commit --amend
```



```
5658BA59463DE3CD,1714573762,First commit
ADC2CDA76C228417,1714573781,No changes from first commit
65961FDC54C99612,1714573805,Quiz1 dropped from main
122B10E156292996,1714574058,Updated quiz2.csv
=0812B2F8AFD496D7,1714574173,Main.csv has quiz1.csv again
=C715F010746496C7,1714582999,Amended prev commit
EAD4A2C47F53665A91E,1714583019,Amended prev commit
```

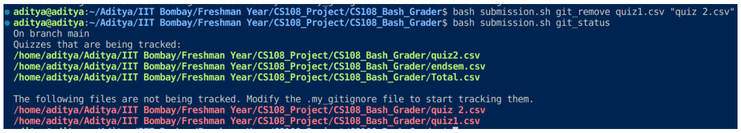
Figure 22: .git_log after amending the last commit

If no commit message is specified, the script gets a commit message from the internet from https://whatthecommit.com/. I am not personally responsible for the language of these prompts. Using the up arrow, w or W, the user can choose to use these commits for his commit message. With a one in five probability, a certain Easter egg appears.

12 git_add and git_remove

These modify the .my_gitignore file to specify certain files to always copy or to always exclude during git_commit. .my_gitignore can handle the same regex expressions that can be handled by bash, as well as the ! command before the regex to indicate not to ignore the file ever. my_gitignore is parsed from top to bottom, so ignoring a file once and then unignoring it will cause it to not be ignored, and vice versa.
To view which files are being tracked:

```
bash submission.sh git_remove quiz1.csv "quiz 2.csv"
bash submission.sh git_status
```



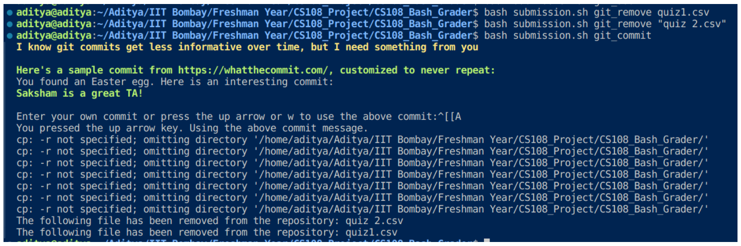
```
aditya@aditya:~/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader$ bash submission.sh git_remove quiz1.csv "quiz 2.csv"
aditya@aditya:~/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader$ bash submission.sh git_status
On branch main
Quizzes that are being tracked:
/home/aditya/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader/quiz2.csv
/home/aditya/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader/endsem.csv
/home/aditya/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader/Total.csv

The following files are not being tracked. Modify the .my_gitignore file to start tracking them.
/home/aditya/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader/quiz 2.csv
/home/aditya/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader/quiz1.csv
```

Figure 23: git_status with quiz1 and quiz 2 not tracked

To add all csv files present in the `WORKING_DIRECTORY`:

```
bash submission.sh git_add .
(or) bash submission.sh git_add *
```

Equivalent commands work for git_remove, or you can equivalently pass the -r flag to git_add to specify removal instead of addition.
The cp errors are expected, since parse_gitignore works by calling:



```
● aditya@aditya:~/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader$ bash submission.sh git_remove quiz1.csv
● aditya@aditya:~/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader$ bash submission.sh git_remove "quiz 2.csv"
● aditya@aditya:~/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader$ bash submission.sh git_commit
  I know git commits get less informative over time, but I need something from you

  Here's a sample commit from https://whatthecommit.com/, customized to never repeat:
  You found an Easter egg. Here is an interesting commit:
  Saksham is a great TA!

  Enter your own commit or press the up arrow or w to use the above commit:^[[A
  You pressed the up arrow key. Using the above commit message.
  cp: -r not specified; omitting directory '/home/aditya/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader/'
  cp: -r not specified; omitting directory '/home/aditya/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader/'
  cp: -r not specified; omitting directory '/home/aditya/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader/'
  cp: -r not specified; omitting directory '/home/aditya/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader/'
  cp: -r not specified; omitting directory '/home/aditya/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader/'
  cp: -r not specified; omitting directory '/home/aditya/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader/'
  cp: -r not specified; omitting directory '/home/aditya/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader/'
  The following file has been removed from the repository: quiz 2.csv
  The following file has been removed from the repository: quiz1.csv
```

Figure 24: Committing after git_remove is called. Note that quiz1.csv and quiz 2.csv are not tracked.

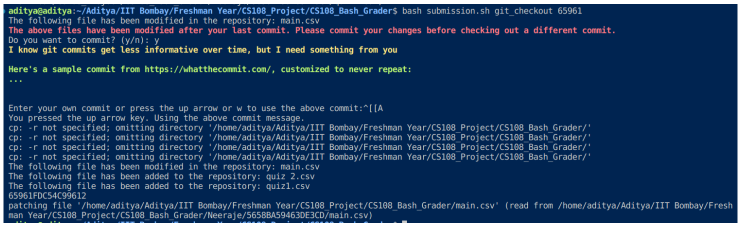
```
selected_quizzes=("${selected_quizzes[@]/$quiz}")
```

which replaces the instance of $quiz in the array with a blank space. These error messages can safely be suppressed, since they try to make cp copy the `WORKING_DIRECTORY`, which fails in the absence of the recursive flag.

13 git_checkout

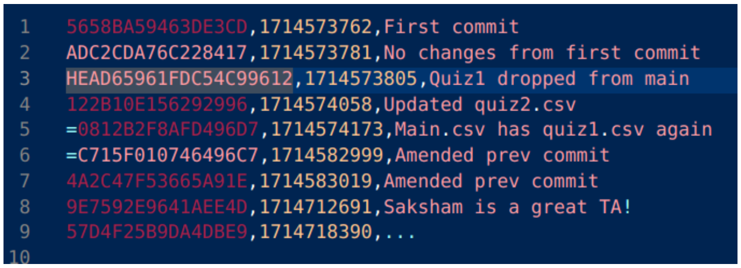
This command checks out the commit whose hash is closest to the query hash passed as an argument. If no existing hash is within a Levenshtein distance of 5 to the query hash, it reports the same.
The query must at least be of length 4.
git_checkout also checks whether the local directory is the same as either the commit you are checking out or the latest two commits in .git_log. If not, it prompts the user to save the local repository as a commit.
The reason for checking out the latest two commits was because otherwise, if you check out a commit A different from the latest commit B, and you call git_checkout main, the script prompts you to store the local directory as commit C, which is chronologically after commit B. If you git_checkout main again, commit C is the latest commit, but then commit B and commit C are different, so B is stored as a new commit D, and so on, creating an infinite loop where git_checkout main never is actually at the tip of main unless the force flag is used.

If there have been changes in the local repository, git_checkout prompts for the new commit to be saved before checking out the commit.
It also tells the user what files have changed, been added or deleted.



```
aditya@aditya:~/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader$ bash submission.sh git_checkout 65961
The following file has been modified in the repository: main.csv
The above files have been modified after your last commit. Please commit your changes before checking out a different commit.
Do you want to commit? (y/n): y
I know git commits get less informative over time, but I need something from you

Here's a sample commit from https://whatthecommit.com/, customized to never repeat:
...


Enter your own commit or press the up arrow or w to use the above commit:^[[A
You pressed the up arrow key. Using the above commit message.
cp: -r not specified; omitting directory '/home/aditya/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader/'
cp: -r not specified; omitting directory '/home/aditya/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader/'
cp: -r not specified; omitting directory '/home/aditya/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader/'
cp: -r not specified; omitting directory '/home/aditya/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader/'
The following file has been modified in the repository: main.csv
The following file has been added to the repository: quiz 2.csv
The following file has been added to the repository: quiz1.csv
65961FDC54C99612
patching file '/home/aditya/Aditya/IIT Bombay/Freshman Year/CS108_Project/CS108_Bash_Grader/main.csv' (read from /home/aditya/Aditya/IIT Bombay/Fresh
man Year/CS108_Project/CS108_Bash_Grader/Neeraje/5658BA59463DE3CD/main.csv)
```

Figure 25: git_checkout of commit 6596 when at commit 122B prompts for the local repository to be stored as a new commit, as the local repository is not equivalent to either of the two most recent commits. The addition of quiz 2.csv and quiz1.csv is because the latest commit in .git_log was committed after git_remove was called, and hence did not have quiz1 or quiz 2.



```
5658BA59463DE3CD,1714573762,First commit
ADC2CDA76C228417,1714573781,No changes from first commit
HEAD65961FDC54C99612,1714573805,Quiz1 dropped from main
122B10E156292996,1714574058,Updated quiz2.csv
=0812B2F8AFD496D7,1714574173,Main.csv has quiz1.csv again
=C715F010746496C7,1714582999,Amended prev commit
4A2C47F53665A91E,1714583019,Amended prev commit
9E7592E9641AEE4D,1714712691,Saksham is a great TA!
57D4F25B9DA4DBE9,1714718390,...
```

Figure 26: git_log after git_checkout. Note that the position of HEAD has changed. Also, the local repository is stored as the new commit with hash 57D4.

14 Grade

If main.csv already has a total column, grade prompts the user on whether to use the same total column to display a scatterplot of the sorted marks, as well as recommended grade cutoffs for AP, AA, AB, BB, BC, DD (assuming all people lower than DD are FRs).
First, a bell-curve type distribution is created by taking the mean, mean + 30/weightage, mean + 20/weightage, etc. where weightage is the 100/maximum_marks. Then, on the sorted list of totals, the biggest jump in the vicinity of these cutoffs is taken to be the final cutoff.
By using different values for the variable WIDTH, I figured out that the best looking graphs came when I searched for the biggest jump in an interval of width $\frac{bucket\_width}{2}$ around the cutoff, where bucket_width is the distance between the current cutoff and the immediately higher cutoff (with the highest cutoff being the number of students).

```
bash submission.sh grade
```

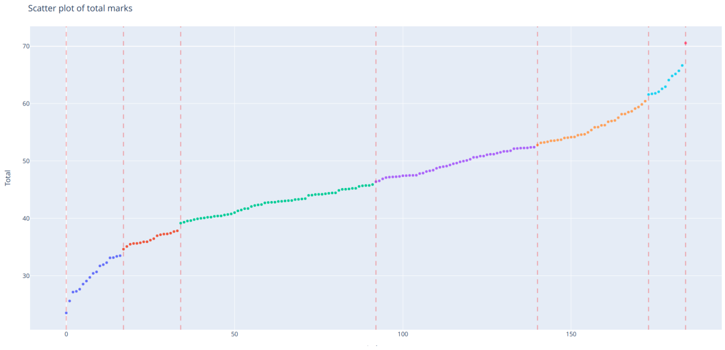


Figure 27: Grade command

15 grade_manual

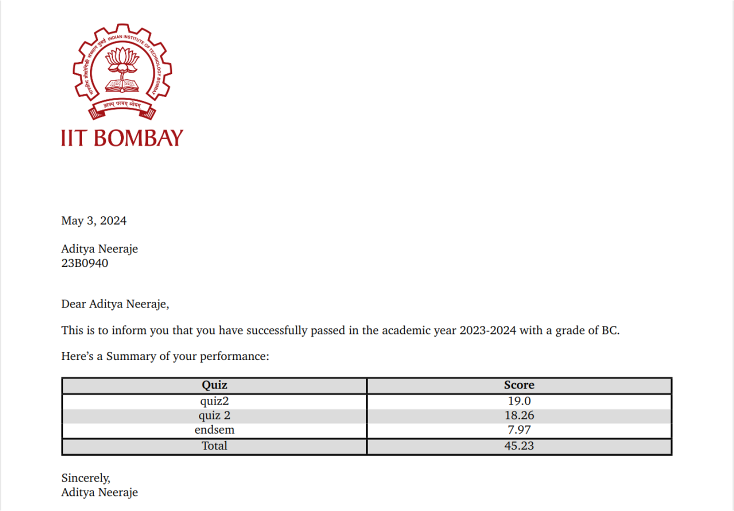
I even have a grade_manual command that allows the user to specify the cutoffs for each grade.
On a desktop with wayland enabled, matplotlib displays the graph, and the cutoff vertical line moves along with the user's cursor and changes color according to the grade whose cutoff it is. The cutoff line does not move more right than the previous cutoff, however, reflecting the fact that the cutoffs must always be in decreasing order.
As with grade, the user is prompted on whether to use a pre-existing Total column, if any.

Both grade and grade_manual generate a main.html file which stores the students' data and grades in tabular form.

16 report_card

Generates a report card for the student whose name or roll number is queried, using a latex template present in the Report_Cards_Template folder.

If no argument is specified, report cards are generated for every student in main.csv. This takes up significant time, so it is not recommended to run this during the viva session unless at the end.



IIT BOMBAY

May 3, 2024

Aditya Neeraje
23B0940

Dear Aditya Neeraje,

This is to inform you that you have successfully passed in the academic year 2023-2024 with a grade of BC.

Here's a Summary of your performance:

Quiz	Score
quiz2	19.0
quiz 2	18.26
endsem	7.97
Total	45.23

Sincerely,
Aditya Neeraje

Figure 28: Report Card for Neeraje

17 git_log

If you are in detached HEAD state, git_log reports that and exits. If not, it prints the hash, commit time and commit message of all the commits until the HEAD is reached. Commits which have been "forgotten" due to being amended later on are not shown. Using the `--oneline` flag, only the commit hash and commit message can be displayed, each commit taking up one line.

18 display_boxplot and display_stripplot

Generic plotly express boxplot and stripplot of the quizzes. If rescale is specified, the plots are all rescaled such that their maximums are at the value passed as an argument to rescale (default is 100 if the `-r--rescale` flag is passed). If a rescale value of 0 is passed, it is ignored. If a non-numeric or negative rescale value is passed, the graph is rescaled to 100.

```
bash submission.sh boxplot --rescale 100
```

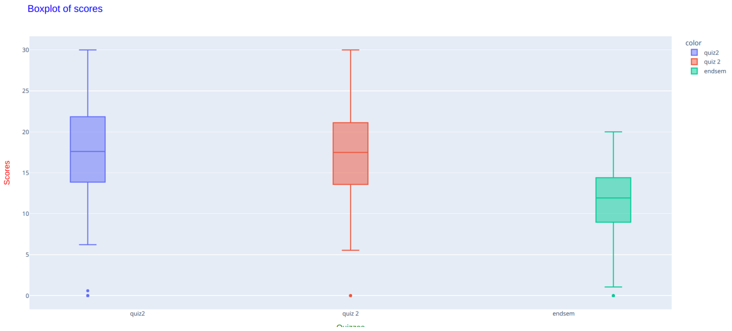


Figure 29: Boxplot of the quizzes

References

[1] URL: `https://github.com/AdityaNeeraje/CS108_Bash_Grader`.