

Evaluation Script

Mike Nguyen

1. export your data to csv (use choice text)
2. change name to mid_semester.csv
3. Import your document by changing the file.path to your directory containing the csv file

```
# library
library(tidyverse)
library(rio)
# import
# this is the only line you need to change
data <- rio::import(file.path(getwd(), "mid_semester.csv")) %>%

  slice(-c(1, 2)) %>% # remove 2 and 3 rows (questions description)

# change members name
rename(
  member1 = Q5_1,
  member2 = Q5_2,
  member3 = Q5_3,
  member4 = Q5_4,
  member5 = Q5_5,
  member6 = Q5_6
) %>% # 6 is rater's own evaluation

# rename other variables
rename(
  name = Q1,
  id = Q2,
  group = Q3,
  overall = Q4,
  comment = Q10
) %>%

# make sure percentage questions are in numeric format
mutate_at(vars(starts_with("Q")), as.numeric) %>%

# take sum across columns
mutate(score_1 = select(., ends_with("_1")) %>% rowSums()) %>%
mutate(score_2 = select(., ends_with("_2")) %>% rowSums()) %>%
mutate(score_3 = select(., ends_with("_3")) %>% rowSums()) %>%
mutate(score_4 = select(., ends_with("_4")) %>% rowSums()) %>%
mutate(score_5 = select(., ends_with("_5")) %>% rowSums()) %>%
mutate(score_6 = select(., ends_with("_6")) %>% rowSums()) %>%

test = data %>%
```

```

# transform data
# select(starts_with("member")| starts_with("score"), name) %>%
pivot_longer(
  cols = starts_with("score"),
  names_to = "order",
  values_to = "evaluation"
) %>%
pivot_longer(
  cols = starts_with("member"),
  names_to = "order1",
  values_to = "member_rated"
) %>%

mutate(order = str_sub(order, 7,8 )) %>%
mutate(order1 = str_sub(order1, 7,8 )) %>%
mutate(match = if_else(order == order1, 1, 0)) %>%
filter(match ==1 ) %>%
select(-c(order, order1, match)) %>%

filter(member_rated != "")

test1= test %>%
  # get group size
  filter(evaluation !=0) %>%
  group_by(group, name) %>%
  summarise(group_size = n()) %>%
  mutate(supposed_contribution = 100/group_size)

test2 = test %>%
  full_join(test1, by = c("name","group")) %>%

  # get how each rater rates their peers on multiple assignments
  mutate(within_rater_evaluation = evaluation/group_size)

test3 = test2 %>%
  # get average score across raters
  group_by(member_rated) %>%
  summarise(final_score = mean(within_rater_evaluation))

test4 = test3 %>%
  full_join(test2, by = ("member_rated")) %>%
  mutate(weight = if_else(final_score >= supposed_contribution,1,final_score/supposed_contribution))

final_table = test4 %>%
  select(member_rated,weight) %>%
  unique()

```

The final_table is the only thing you need to care