HW 2

Iris Zhong

2022-10-09

This dataset comes from a free online tutoring platform – ASSISTments. It is called a “tutoring” system, because if students get a question wrong, they are provided with a small “tutoring” session where they must answer a few questions that break the problem down into steps. The students in this dataset were all middle school students.

Data description: <https://sites.google.com/site/assistmentsdata/datasets/2012-13-school-data-with-affect?authuser=0>

Data codebook: <https://sites.google.com/site/assistmentsdata/an-explanation-on-how-to-interpret-our-data-sets?authuser=0>

Citation: Wang, Y., Heffernan, N, & Heffernan, C. (2015) Towards better affect detectors: effect of missing skills, class features and common wrong answers. Proceedings of the Fifth International Conference on Learning Analytics And Knowledge. pp 31-35.

Feng, M., Heffernan, N.T., & Koedinger, K.R. (2009). Addressing the assessment challenge in an Intelligent Tutoring System that tutors as it assesses. The Journal of User Modeling and User-Adapted Interaction.19, 243-266. (Based on CP15)

# Data loading

Load the data directly from google drive.

drive\_deauth() # avoid google authorization  
temp <- tempfile(fileext = ".zip")  
dl <- drive\_download(  
 as\_id("1cU6Ft4R3hLqA7G1rIGArVfelSZvc6RxY"), path = temp, overwrite = TRUE)

## File downloaded:

## • '2012-2013-data-with-predictions-4-final.zip'  
## <id: 1cU6Ft4R3hLqA7G1rIGArVfelSZvc6RxY>

## Saved locally as:

## • '/var/folders/3b/vw5330t962s29tr9xkbnf3gh0000gn/T//RtmpUayPu6/filed5937338fae4.zip'

out <- unzip(temp, exdir = tempdir())  
data <- fread(out, sep = ",")

# Data codebook

Below is the direct quotation of the variable descriptions from the website.

| Variable | Description |
| --- | --- |
| problem\_log\_id | Unique ID of the logged actions. Problem\_log is the table that the biggest in ASSISTments. About 10 millions problems were solved in 2012, so there are 10 million rows in the data base. For each problem, there might be a few attempts that a child made and a few hint requests. We call them actions. We don’t store actions in their own table but if we did it would be bigger as every problem has at least one action (if its correct) and more if the student was incorrect. We store in the “actions” field of problem\_log the actions with time stamps. |
| skill | Skill name associated with the problem (different skills are in different rows). |
| problem\_id | The ID of the problem. |
| user\_id | The ID of the student doing the problem. |
| assignment\_id | Two different assignments can have the same sequence id. Each assignment is specific to a single teacher/class. |
| assistment\_id | The ID of the ASSISTment. An ASSISTment consists of one or more problems. |
| start\_time | Timestamp when the problem starts. |
| end\_time | Timestamp when the problem ends. |
| problem\_type | choose\_1: Multiple choice (radio buttons)  algebra: Math evaluated string (text box)  fill\_in: Simple string-compared answer (text box)  open\_response: Records student answer, but their response is always marked correct  *more problem\_type in the data: choose\_n, rank* |
| original | 1 = Main problem  0 = Scaffolding problem |
| correct | 1 = Correct on first attempt  decimal values are calculated as a partial credit based on the number of hints and attempts needed to solve (based on teacher setting)  0 = student either saw the answer, exhausted partial credit from too many hints/attempts, or (based on teacher setting) answered incorrectly on the first attempt  When observed as a dependent variable, it is recommended that this value be converted to a binary variable using the formula: 1 = correct, <1 = incorrect |
| bottom\_hint | Whether or not the student asks for all hints. |
| hint\_count | Number of hints on this problem. |
| actions | Every action on this problem. |
| attempt\_count | Number of student attempts on this problem. |
| ms\_first\_response | The time in milliseconds for the student’s first response. |
| tutor\_mode | tutor, test mode, pre-test, or post-test  *only tutor and test in the data* |
| sequence\_id | The content id of the problem set. Different assignments that are assigned the same problem set will have the same sequence id. Again the terminology is confused as years ago when ASSISTments was starting we called problem sets sequences. But a problem set in our modern use of the term is really stored as a sequence. Most sequences are simple, but it’s possible to build a problem set that is a hierarchical tree of problem sets. |
| student\_class\_id | The class ID. |
| position | Assignment position on the class assignments page. |
| type | This is the type of the head section of the problem set. Each problem set is usually one of the following three.  Linear - Student completes all problems in a predetermined order.  Random - Student completes all problems, but each student is presented with the problems in a different random order.  Mastery - Random order, and student must “master” the problem set by getting a certain number of questions correct in a row before being able to continue. ASSISTments calls problem sets that have a head section that is of type mastery a “Skill Builder”.  *more types in the data: ChooseCondition, NumericLimit, RandomIterate* |
| base\_sequence\_id | This is to account for if a sequence has been copied. This will point to the original copy, or be the same as sequence\_id if it hasn’t been copied. |
| skill\_id | ID of the skill associated with the problem (different skills are in different rows). |
| teacher\_id | The ID of the teacher who assigned the problem. |
| school\_id | The ID of the school where the problem was assigned. |
| overlap\_time | The time in milliseconds for the student’s overlap time. |
| template\_id | The template ID of the ASSISTments. ASSISTments with the same template ID have similar questions. |
| answer\_id | The answer ID for multi-choice questions. |
| answer\_text | The answer text for fill-in questions. |
| first\_action | The type of first action: attempt or ask for a hint. |
| problemlog\_id | Unique ID of the logged actions. |
| Average\_confidence(FRUSTRATED) | Predicted Frustration of student for the problem. Value close to “0” being less frustrated and close to “1” being more frustrated. |
| Average\_confidence(CONFUSED) | Predicted Confusion of student for the problem. Value close to “0” being less confused and close to “1” being more confused. |
| Average\_confidence(CONCENTRATING) | Predicted Engaged Concentration of student for the problem. Value close to “0” being less concentrated and “1” being more concentrated. |
| Average\_confidence(BORED) | Predicted Boredom of student for the problem. Value close to”0” being less bored and “1” being more bored. |
|  |  |

# Data cleaning

The data is already in a tidy format. For this part, I’ll only remove columns that I won’t use, and rename a few variables.

data <- data %>%  
 dplyr::select(-c(actions, overlap\_time, problemlogid, answer\_id, answer\_text)) %>%  
 rename(frustrated\_confidence = `Average\_confidence(FRUSTRATED)`,  
 confused\_confidence = `Average\_confidence(CONFUSED)`,  
 concentrating\_confidence = `Average\_confidence(CONCENTRATING)`,  
 bored\_confidence = `Average\_confidence(BORED)`)

# Data summary

Check the number of distinct values for each variable:

data %>% summarize\_all(n\_distinct)

## problem\_log\_id skill problem\_id user\_id assignment\_id assistment\_id  
## 1 6123270 199 179999 46674 189760 138555  
## start\_time end\_time problem\_type original correct bottom\_hint hint\_count  
## 1 4672374 6121336 6 2 13 3 15  
## attempt\_count ms\_first\_response tutor\_mode sequence\_id student\_class\_id  
## 1 30 362600 2 112403 2661  
## position type base\_sequence\_id skill\_id teacher\_id school\_id template\_id  
## 1 6747 7 110648 266 1331 662 96403  
## first\_action frustrated\_confidence confused\_confidence  
## 1 3 831 2476  
## concentrating\_confidence bored\_confidence  
## 1 842 11213

Check the summary statistics of the continuous variables:

data %>%  
 select(hint\_count, attempt\_count,   
 frustrated\_confidence, confused\_confidence, concentrating\_confidence, bored\_confidence) %>%  
 describe()

## vars n mean sd min max range se  
## hint\_count 1 6123270 0.34 0.99 0.00 14.00 14.00 0  
## attempt\_count 2 6123270 1.34 1.06 0.00 29.00 29.00 0  
## frustrated\_confidence 3 6123270 0.39 0.10 0.36 0.87 0.51 0  
## confused\_confidence 4 6123270 0.04 0.19 0.00 1.00 1.00 0  
## concentrating\_confidence 5 6123270 0.68 0.17 0.17 0.77 0.60 0  
## bored\_confidence 6 6123270 0.26 0.29 0.00 1.00 1.00 0

Check the counts of categorical variables:

count(data, correct)

## correct n  
## 1: 0.000 1976383  
## 2: 0.250 1211  
## 3: 0.375 1  
## 4: 0.500 2395  
## 5: 0.600 24  
## 6: 0.625 4  
## 7: 0.650 10  
## 8: 0.750 1629  
## 9: 0.850 10  
## 10: 0.875 6  
## 11: 0.950 9  
## 12: 0.975 24  
## 13: 1.000 4141564

count(data, problem\_type)

## problem\_type n  
## 1: algebra 3500688  
## 2: choose\_1 1847657  
## 3: choose\_n 11597  
## 4: fill\_in\_1 742960  
## 5: open\_response 17642  
## 6: rank 2726

count(data, original)

## original n  
## 1: 0 303533  
## 2: 1 5819737

count(data, bottom\_hint)

## bottom\_hint n  
## 1: 0 5335070  
## 2: 1 727852  
## 3: NA 60348

count(data, tutor\_mode)

## tutor\_mode n  
## 1: test 2272  
## 2: tutor 6120998

count(data, type)

## type n  
## 1: ChooseConditionSection 3489  
## 2: LinearSection 3349671  
## 3: MasterySection 1868746  
## 4: NumericLimitSection 16  
## 5: PlacementsSection 3947  
## 6: RandomChildOrderSection 875949  
## 7: RandomIterateSection 21452

count(data, first\_action)

## first\_action n  
## 1: 0 5782418  
## 2: 1 305009  
## 3: 2 35843

bottom\_hint contains NA, and first\_action contains three levels. Since these values are not clear enough, these two variables should probably be removed.

In addition, almost all of the response data is binary (0 or 1). As suggested by the code book, recode all values in correct that are less than 1 to 0.

data <- data %>%  
 dplyr::select(-c(bottom\_hint, first\_action)) %>%  
 mutate(correct = ifelse(correct == 1, 1, 0))

To understand the time frame of the data: first, check the earliest and latest log in the data.

min(data$start\_time)

## [1] "2012-09-01 00:00:06 UTC"

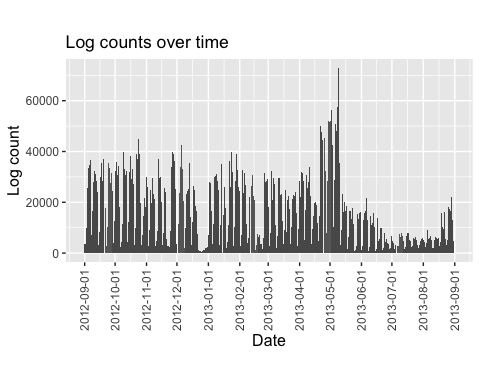
max(data$start\_time)

## [1] "2013-08-31 23:59:57 UTC"

The earliest start time is 2012-09-01, and the latest start time is 2013-08-31.

The time distribution of the logs:

ggplot(data,   
 aes(x=as.Date(start\_time))) +   
 geom\_bar() +   
 scale\_x\_date(date\_breaks="1 month") +   
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust=1),   
 axis.title=element\_text(size=12), axis.text=element\_text(size=9),   
 aspect.ratio=1/2) +  
 labs(title = 'Log counts over time', x = "Date", y = "Log count")



ASSISTments was used the most in May, the end of the school year. Its usage was low during the holidays (winter and summer vacations). There was a sizable variation of log counts within months. I suspect that days with very low usages are weekends, when fewer assignments were due.

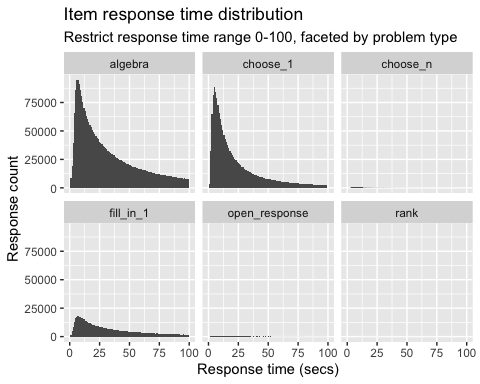
Item response time distribution:

data <- data %>%  
 mutate(response\_time = difftime(end\_time, start\_time, units = "secs"))

ggplot(data, aes(x = as.numeric(response\_time))) +  
 geom\_histogram(binwidth = 1) +  
 xlim(0, 100) +  
 labs(x = "Response time (secs)", y = "Response count",  
 title = "Item response time distribution",  
 subtitle = "Restrict response time range 0-100, faceted by problem type") +  
 facet\_wrap(~problem\_type)

## Warning: Removed 1009282 rows containing non-finite values (stat\_bin).

## Warning: Removed 12 rows containing missing values (geom\_bar).



favstats(data$response\_time)

## Warning in fav\_stats(x, ..., na.rm = na.rm): Auto-converting difftime to  
## numeric.

## min Q1 median Q3 max mean sd n missing  
## -23667870 11.142 26.917 67.181 26084890 4454.746 141699.7 6123270 0

# restrict summary statistics to be 0 - 1000s  
favstats(data[data$response\_time > 0 & data$response\_time < 1000,]$response\_time)

## Warning in fav\_stats(x, ..., na.rm = na.rm): Auto-converting difftime to  
## numeric.

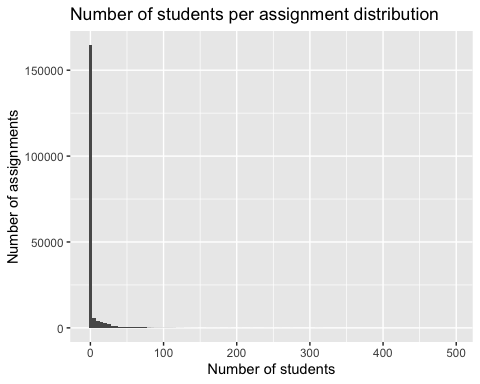
## min Q1 median Q3 max mean sd n missing  
## 0.01215911 11.077 26.623 65.814 999.992 57.70174 85.88074 6082650 0

Most items were answered within 1 minute. There were extreme outliers on both sides of the distribution. From the plots, the response time among problem types was not too different.

# Data exploration

Some more data exploration. First question: how many students work on the same assignment/assistment?

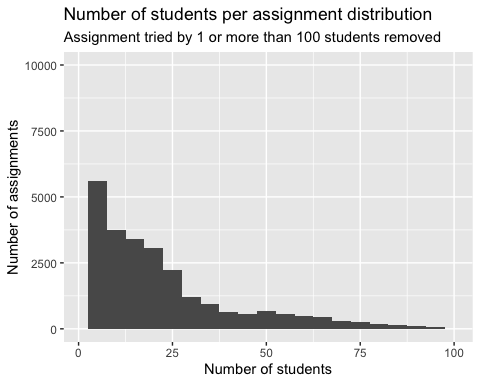
data %>%   
 group\_by(assignment\_id) %>%   
 summarize(n\_student = uniqueN(user\_id)) %>%  
 ggplot(., aes(x = n\_student)) +  
 geom\_histogram(binwidth = 5) +  
 labs(title = "Number of students per assignment distribution",  
 x = "Number of students",  
 y = "Number of assignments")



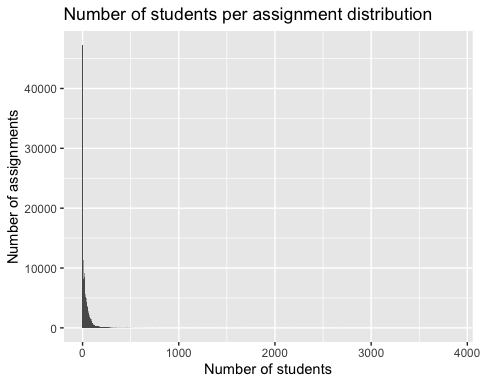
data %>%   
 group\_by(assignment\_id) %>%   
 summarize(n\_student = uniqueN(user\_id)) %>%  
 ggplot(., aes(x = n\_student)) +  
 geom\_histogram(binwidth = 5) +  
 xlim(1, 100) +  
 ylim(0, 10000) +   
 labs(title = "Number of students per assignment distribution",  
 subtitle = "Assignment tried by 1 or more than 100 students removed",  
 x = "Number of students",  
 y = "Number of assignments")

## Warning: Removed 533 rows containing non-finite values (stat\_bin).

## Warning: Removed 2 rows containing missing values (geom\_bar).



data %>%   
 group\_by(assistment\_id) %>%   
 summarize(n\_student = uniqueN(user\_id)) %>%  
 ggplot(., aes(x = n\_student)) +  
 geom\_histogram(binwidth = 5) +  
 labs(title = "Number of students per assignment distribution",  
 x = "Number of students",  
 y = "Number of assignments")



Most assignments were only finished by one student, similarly with assistments.

Second questions: how many students in a class?

data %>%  
 group\_by(student\_class\_id) %>%  
 summarize(n\_student = uniqueN(user\_id)) %>%  
 describe()

## vars n mean sd median trimmed mad min max  
## student\_class\_id 1 2661 24143.26 2070.68 24273 24323.58 1859.18 11393 27386  
## n\_student 2 2661 23.89 37.63 14 16.42 19.27 1 516  
## range skew kurtosis se  
## student\_class\_id 15993 -1.50 4.87 40.14  
## n\_student 515 4.94 39.99 0.73

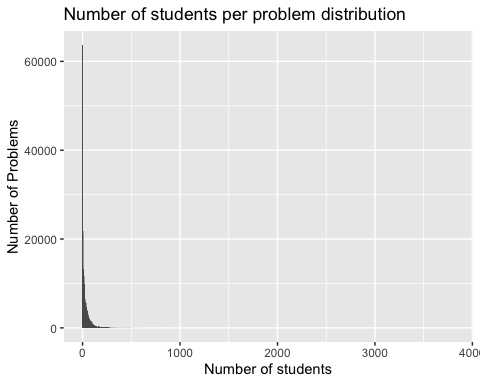
On average, there were 24 students in a class. The median value was 14.

Third question: How many users tried the same problem?

data %>%   
 group\_by(problem\_id) %>%   
 summarize(n\_student = uniqueN(user\_id)) %>%  
 describe()

## vars n mean sd median trimmed mad min  
## problem\_id 1 179999 493491.34 231562.88 578837 515790.42 201486.82 1  
## n\_student 2 179999 32.75 106.74 9 17.02 11.86 1  
## max range skew kurtosis se  
## problem\_id 767143 767142 -0.74 -0.89 545.80  
## n\_student 3809 3808 17.54 431.91 0.25

data %>%   
 group\_by(problem\_id) %>%   
 summarize(n\_student = uniqueN(user\_id)) %>%  
 ggplot(., aes(x = n\_student)) +  
 geom\_histogram(binwidth = 5) +  
 labs(title = "Number of students per problem distribution",  
 x = "Number of students",  
 y = "Number of Problems")



Similarly, a majority of problems were only attempted by one student.

Fourth question: how many assignments and how many problems did a student try?

data %>%   
 group\_by(user\_id) %>%   
 summarize(assignment\_id = uniqueN(assignment\_id),  
 n\_problem = uniqueN(problem\_id)) %>%  
 describe()

## vars n mean sd median trimmed mad min  
## user\_id 1 46674 192188.91 26594.03 195778.5 195725.53 24342.81 21421  
## assignment\_id 2 46674 17.23 31.35 6.0 9.38 7.41 1  
## n\_problem 3 46674 126.30 221.18 48.0 77.35 57.82 1  
## max range skew kurtosis se  
## user\_id 228213 206792 -1.31 2.14 123.10  
## assignment\_id 337 336 3.56 15.35 0.15  
## n\_problem 4698 4697 4.47 30.81 1.02

On average, a student would finish 17 assignments and 126 problems. The median values were 6 and 48, respectively.

# Research question

Do students who regularly sought hints have a different growth trajectory than students who didn’t like to use hints?