Exploring Educational Datasets

INFO 5200 Learning Analytics: Week 2 Homework

Cole Walsh, 4399966

Introducing the Data Context

For this homework, you will be analyzing two public datasets obtained from PSLC DataShop.

- The first dataset provides question-level data of students practicing math problems in academic year 2004-2005 using the Assisstments platform. On this platform, students can attempt a problem many times to get it right and they can ask for more and more hints on a problem until the final hint tells them what the answer is.
- The second dataset derives from an educational game for improving math sense. The online game site is not live anymore but here is a video of the game. Watch it to better understand the variables in the dataset. A special feature of this dataset is that it records data for a randomized experiment with two conditions to test a research hypothesis about how to teach kids "that fractions represent magnitudes of the same basic type as whole numbers."

Loading Datasets

Before you can load data, you need to figure out the format that it is saved in. The file extension typically corresponds to the format, but this is not always the case. R has functions to load all common data files, most of these functions start with `read`, e.g. `read.csv()` for CSVs or `read.tsv()` for tab-separated values. The **foreign** package adds functions to import many additional data file types. For large data files, consider using the `fread` function in the **data.table** package: it's fast and reliable.

The `readRDS()` and `saveRDS()` functions allow you to important and export any object in R. This can be a scala, vector, matrix, data.frame, function, or any other object. Moreover, saving a dataset as an RDS file is much more efficient (smaller file size) than saving it as a CSV.

- Load the Assistments dataset (*info5200.2.assisstments.rds*) into R and call it `asm`.
- Load the fraction game dataset (*info5200.2.gamedata.csv*) into R and call it `fr`.

```r
# add code here to load files
asm <- readRDS('info5200.2.assisstments.rds')
fr <- read.csv('info5200.2.gamedata.csv')
```

Exploring the Assisstments Dataset

It is hard to overstate the importance of understanding the data you are working with. You want to understand the data-generating process, how exactly the data came about. But first, you need to understand what is in the dataset. Look at the first few lines using `head()`.

```r
head(asm)
```

```
##   studentID itemid correctonfirstattempt attempts hints seconds
## 1       136     90                     1        1     0      58
## 2       136     91                     0        1     3      91
## 3       136     92                     1        1     0      11
## 4       136     93                     1        1     0      10
## 5       136     94                     0        2     0      43
## 6       136     95                     1        1     0      13
##                full_start_time             full_finish_time start_day
## 1 01-OCT-04 07.44.43.000000 AM 01-OCT-04 07.45.41.000000 AM 01-OCT-04
## 2 10-DEC-04 09.27.20.000000 AM 10-DEC-04 09.28.51.000000 AM 10-DEC-04
## 3 10-DEC-04 09.28.51.000000 AM 10-DEC-04 09.29.02.000000 AM 10-DEC-04
## 4 10-DEC-04 09.29.02.000000 AM 10-DEC-04 09.29.12.000000 AM 10-DEC-04
## 5 10-DEC-04 09.29.12.000000 AM 10-DEC-04 09.29.55.000000 AM 10-DEC-04
## 6 15-OCT-04 07.44.14.000000 AM 15-OCT-04 07.44.27.000000 AM 15-OCT-04
##   start_time finish_day finish_time
## 1    7.44.43  01-OCT-04     7.45.41
## 2    9.27.20  10-DEC-04     9.28.51
## 3    9.28.51  10-DEC-04      9.29.2
## 4     9.29.2  10-DEC-04     9.29.12
## 5    9.29.12  10-DEC-04     9.29.55
## 6    7.44.14  15-OCT-04     7.44.27
```

Based on the first few lines of data, and what we know about the dataset, we can infer the following:

- *studentID* is an identifier for students
- *itemid* is an identifier for math questions
- *correctonfirstattempt* is an indicator of whether a student answered correctly on the first attempt
- *attempts* is the number of answer attempts required
- *hints* the number of hints a student requested
- *seconds* time spent on the question in seconds
- the remaining columns provide start and end times and dates for each question

It also shows us that the dataset is in **long format** (1 row = 1 event) instead of wide format (1 row = 1 individual). However, as you can see from the *attempts* variable, you do not have data on each attempt, but a question-level rollup. The data is at the student-question level, which means that there is one row for each question a student attempted that summarizes interaction with the question (performance indicators and time spent).

Now answer the following questions with this dataset.

Q1: How many unique individuals are in there?

```r
length(unique(asm$studentID))
```

```
## [1] 912
```

There are 912 unique indivduals in the dataset.

Q2: How many unique questions are there?

```r
length(unique(asm$itemid))
```

```
## [1] 1709
```

There are 1709 unique questions in the dataset.

Q3: What is the rate of getting it right on the first attempt?

```
mean(asm$correctonfirstattempt)
```

```
## [1] 0.4047563
```

Individuals get it right on the first attempt approximately 40.5% of the time.

Q4: What is the rate of asking for hints?

```
mean(asm$hints)
```

```
## [1] 0.7563059
```

Individuals ask for ~0.756 hints per question on average.

Q5: How long do students spend on a question on average?

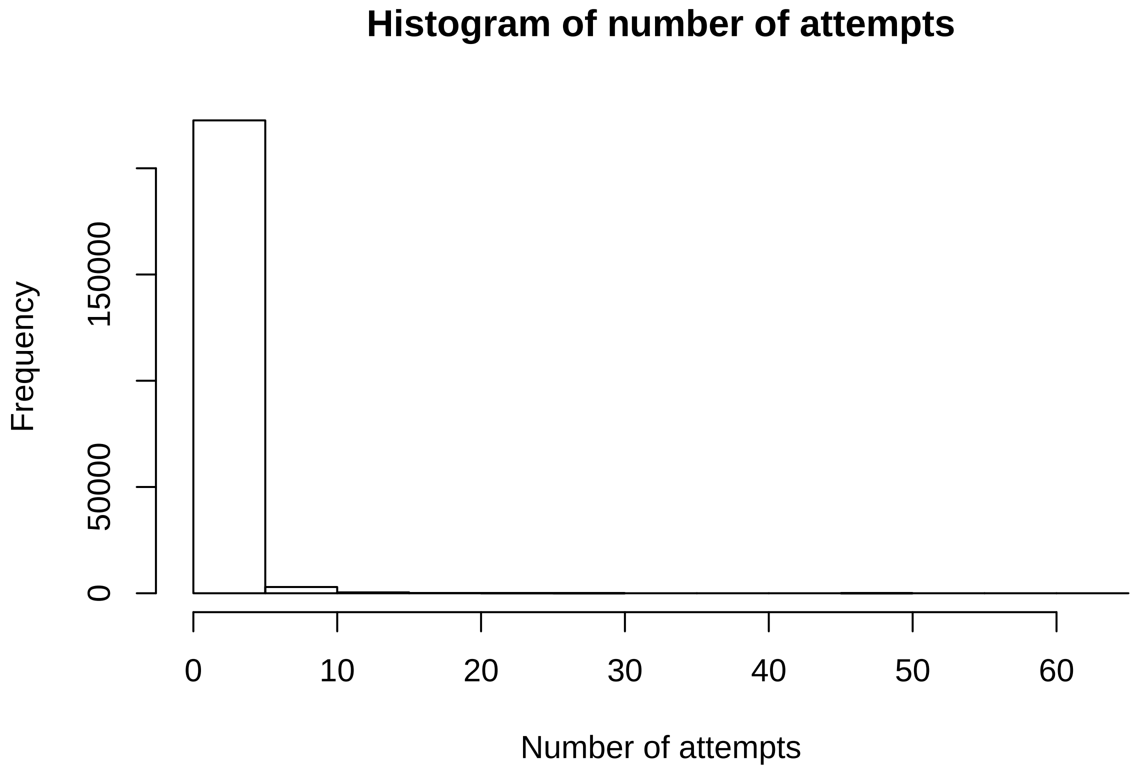
```
mean(asm$seconds)
```

```
## [1] 48.65293
```

Individuals spend approximately 48.7 seconds on each question, on average.

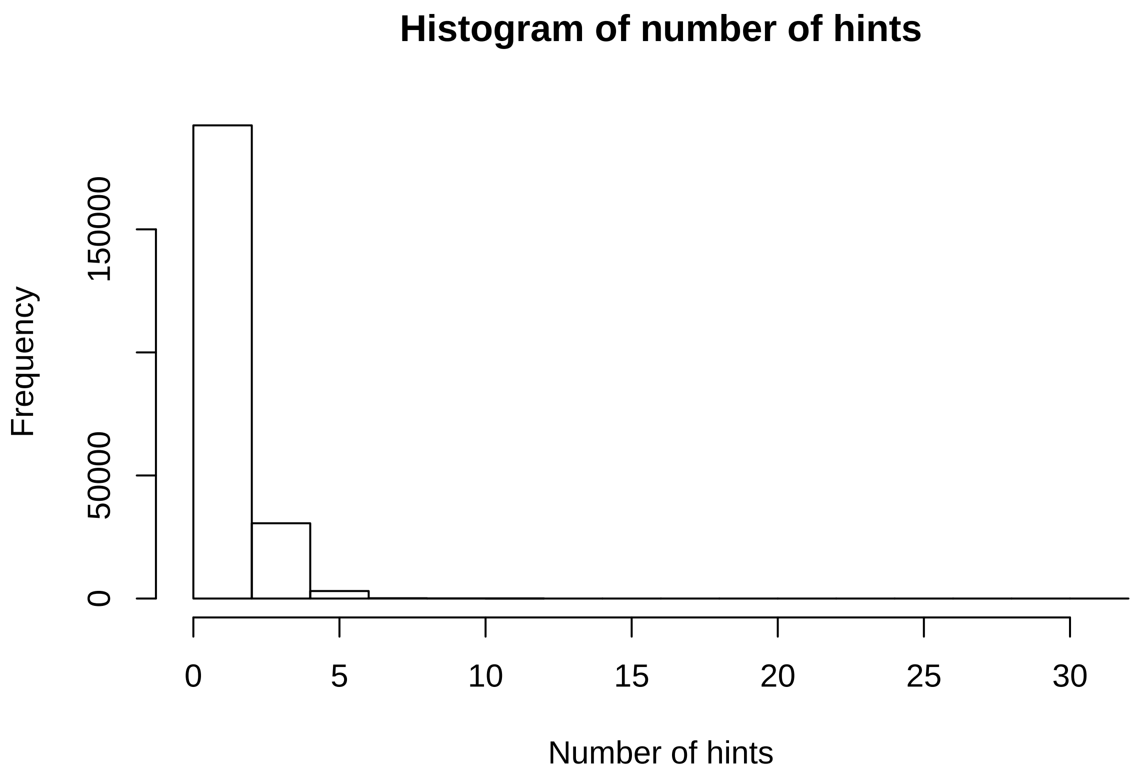
Q6: Plot the distribution of attempts as a histogram:

```
hist(asm$attempts, main = 'Histogram of number of attempts', xlab = 'Number of attempts')
```



Q7: Plot the distribution of hints as a histogram:

```
hist(asm$hints, main = 'Histogram of number of hints', xlab = 'Number of hints')
```



Q8: What are the three pair-wise correlations between seconds, attempts, and hints?

```
cor(asm[, c('seconds', 'attempts', 'hints')])
```

```
##            seconds  attempts     hints
## seconds  1.0000000 0.4345258 0.1709763
## attempts 0.4345258 1.0000000 0.1275844
## hints    0.1709763 0.1275844 1.0000000
```

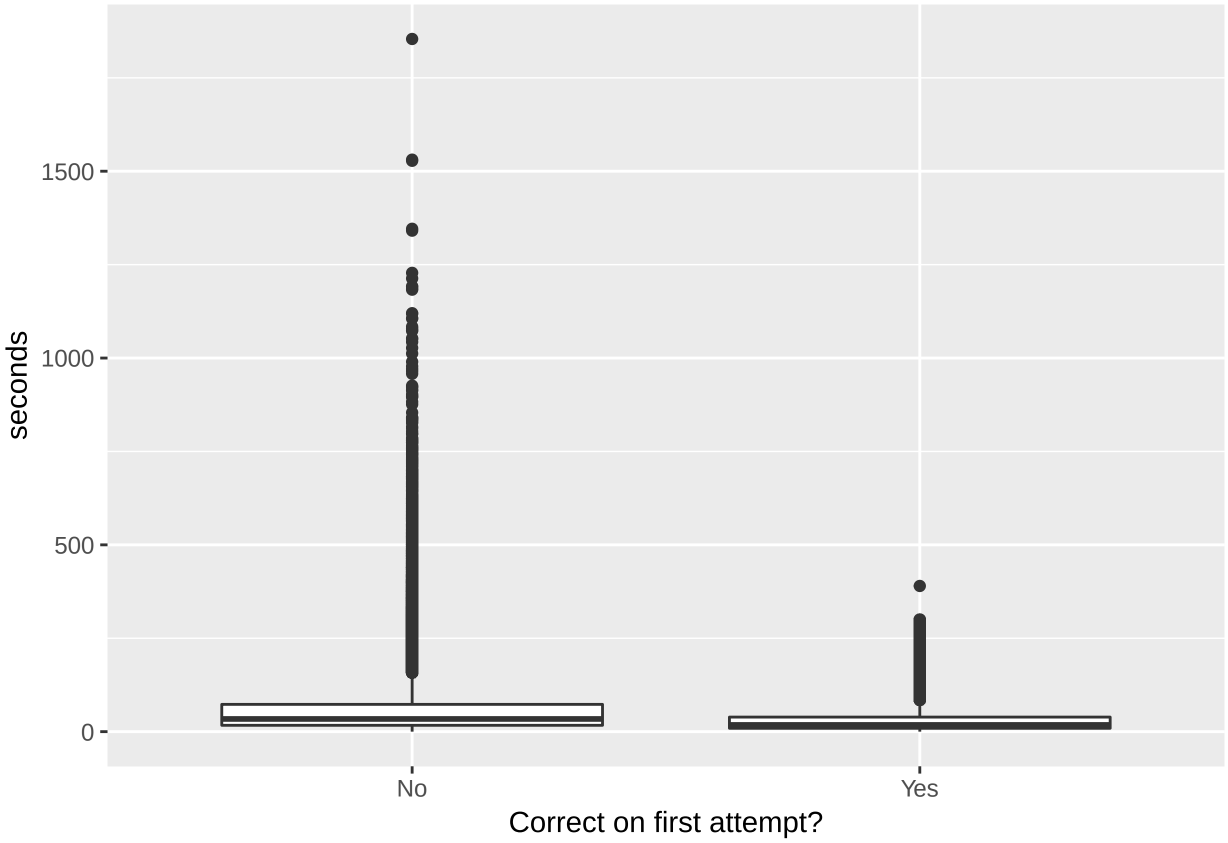
The pearson correlation coefficients are: ~0.435 between seconds and attempts, ~0.171 between seconds and hints, and ~ 0.128 between attempts and hints.

Q9: Plot the distributions of time spent comparing questions that students got right on the first attempts and those where it took more attempts using a boxplot:

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.5.1
```

```
ggplot(asm, aes(x = factor(correctonfirstattempt), y = seconds)) +
  geom_boxplot() +
  xlab('Correct on first attempt?') +
  scale_x_discrete(labels = c('No', 'Yes'))
```



Q10: Tabulate the frequency distribution of hints using `table()`:

```
table(asm$hints)
```

```
## 
##      0      1      2      3      4      5      6      7      8      9 
## 152270  25841  14139  23668   6916   2088    953     74     57     36 
##     10     11     12     13     14     15     16     17     18     19 
##     17      5     16      3      2      3      2      1      2      1 
##     20     21     24     28     31 
##      1      1      1      1      1
```

Q11: Tabulate the frequency distribution of hints against getting it right on the first attempt (note in the output that 6+2+2 handful of students asked for hints before making an attempt and then got it right on their first attempt):

```
table(asm$hints, asm$correctonfirstattempt)
```

```
## 
```

```
##          0     1
##   0  60765 91505
##   1  25835     6
##   2  14137     2
##   3  23666     2
##   4   6916     0
##   5   2088     0
##   6    953     0
##   7     74     0
##   8     57     0
##   9     36     0
##   10    17     0
##   11     5     0
##   12    16     0
##   13     3     0
##   14     2     0
##   15     3     0
##   16     2     0
##   17     1     0
##   18     2     0
##   19     1     0
##   20     1     0
##   21     1     0
##   24     1     0
##   28     1     0
##   31     1     0
```

Q12: Plot the distribution of how many questions students attempted:

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.5.1
```

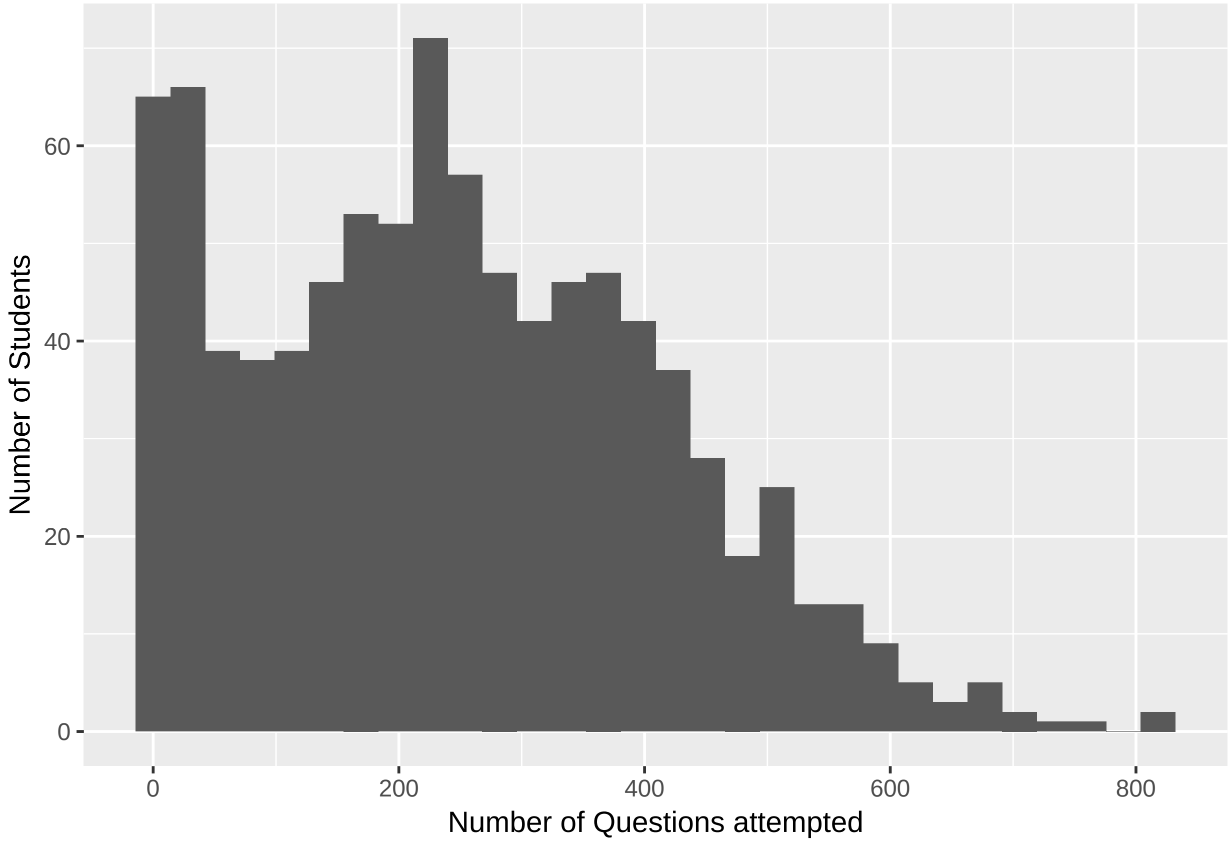
```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
asm %>%
  group_by(studentID) %>%
  summarize(NumQuestions = n()) %>%
  ggplot(., aes(x = NumQuestions)) +
  geom_histogram() +
  xlab('Number of Questions attempted') +
  ylab('Number of Students')
```

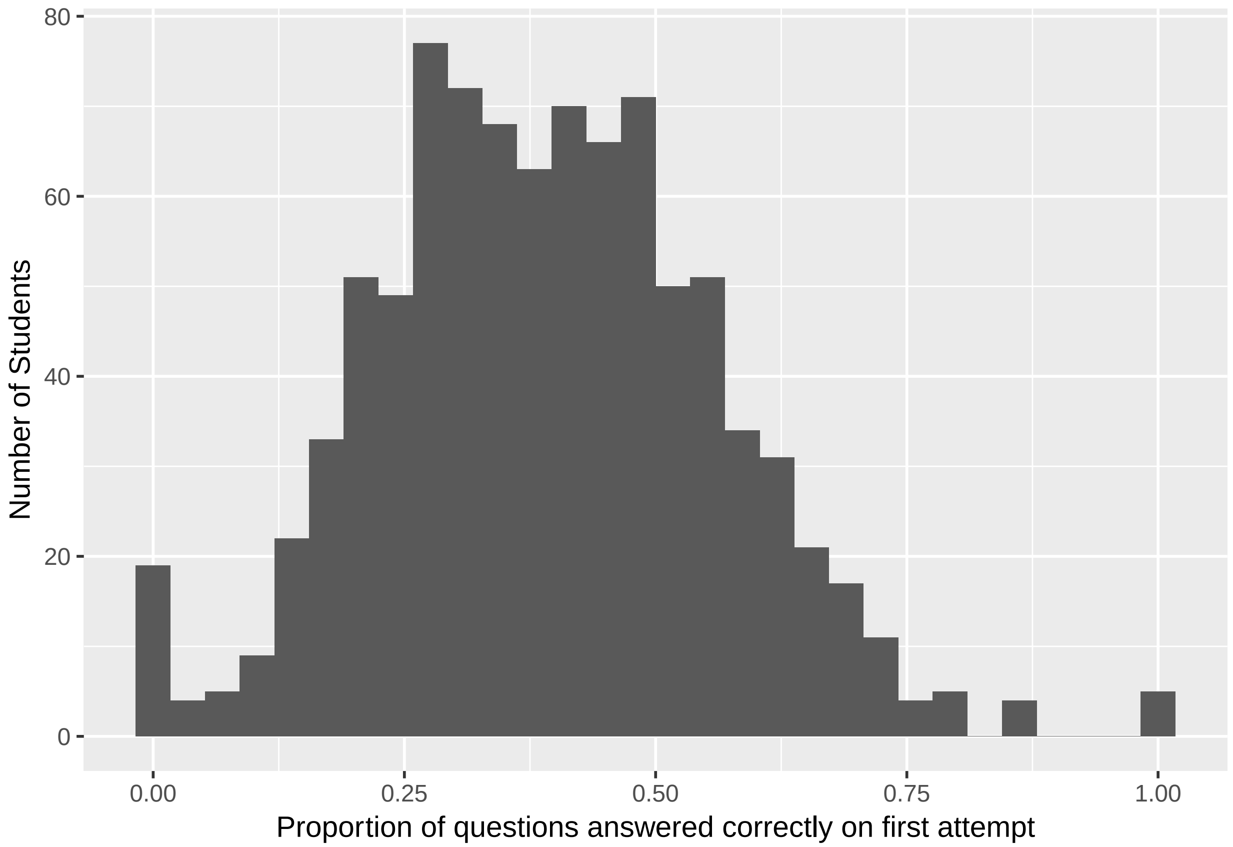
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Q13: Plot the student-level distribution (i.e. 1 value per student) of answering correctly on the first attempt (hint: you first need to compute the proportion of questions that each student got right on first attempt; there are several ways to do this, e.g. using `sapply()`, or loading the `tidyverse` package and using `group_by` and `summarise`, or using syntax from the `data.table` package which is the fastest option; do NOT use a *for* loop unless you really cannot solve it otherwise):

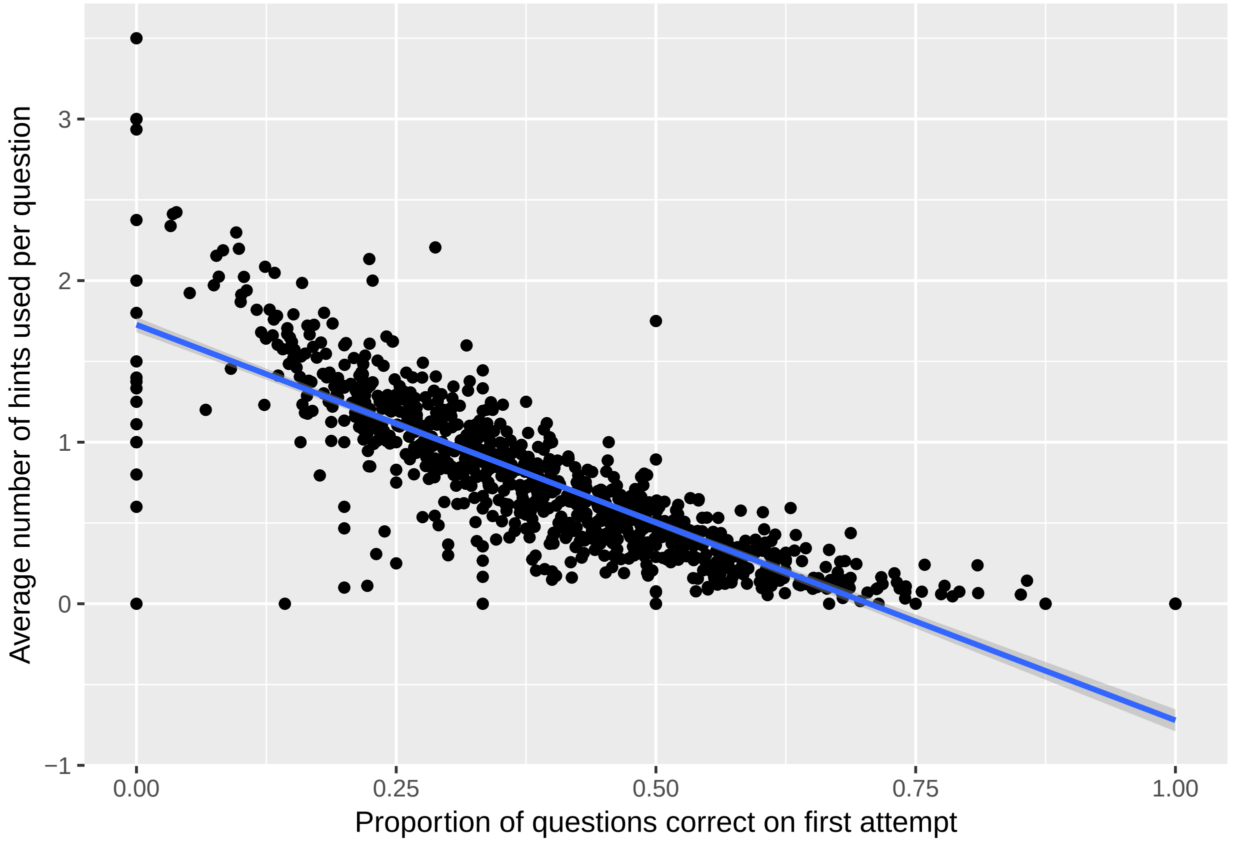
```
asm %>%
  group_by(studentID) %>%
  summarize(PropCorrectFirstAttempt = mean(correctonfirstattempt)) %>%
  ggplot(., aes(x = PropCorrectFirstAttempt)) +
  geom_histogram() +
  xlab('Proportion of questions answered correctly on first attempt') +
  ylab('Number of Students')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Q14: Plot the student-level relationship between getting questions correct (as in Q13; x-axis) and the average number of hints (y-axis) using a scatter plot. Try adding a straight line to fit the data.

```r
asm %>%
  group_by(studentID) %>%
  summarize(PropCorrectFirstAttempt = mean(correctonfirstattempt),
            AverageNumHints = mean(hints)) %>%
  ggplot(., aes(x = PropCorrectFirstAttempt, y = AverageNumHints)) +
  geom_point() +
  geom_smooth(method = 'lm') +
  xlab('Proportion of questions correct on first attempt') +
  ylab('Average number of hints used per question')
```



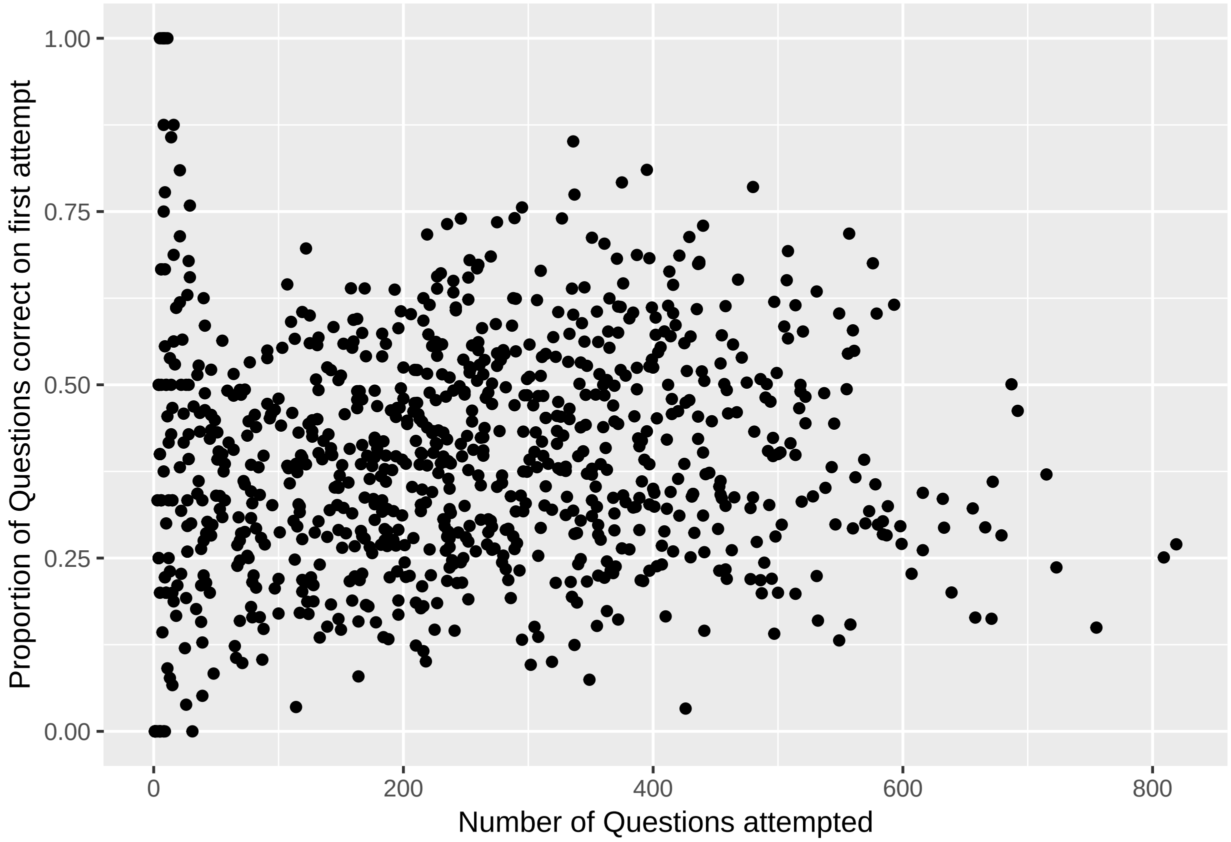
Q15: Are students who attempt more questions (i.e. get more practice) more likely to answer correctly on the first attempt? Provide a correlation and scatterplot.

```r
asm2 <- asm %>%
  group_by(studentID) %>%
  summarize(NumQuestions = n(),
            PropCorrectFirstAttempt = mean(correctonfirstattempt))

cor(asm2$NumQuestions, asm2$PropCorrectFirstAttempt)
```

```
## [1] 0.1007781
```

```r
ggplot(asm2, aes(x = NumQuestions, y = PropCorrectFirstAttempt)) +
  geom_point() +
  xlab('Number of Questions attempted') +
  ylab('Proportion of Questions correct on first attempt')
```

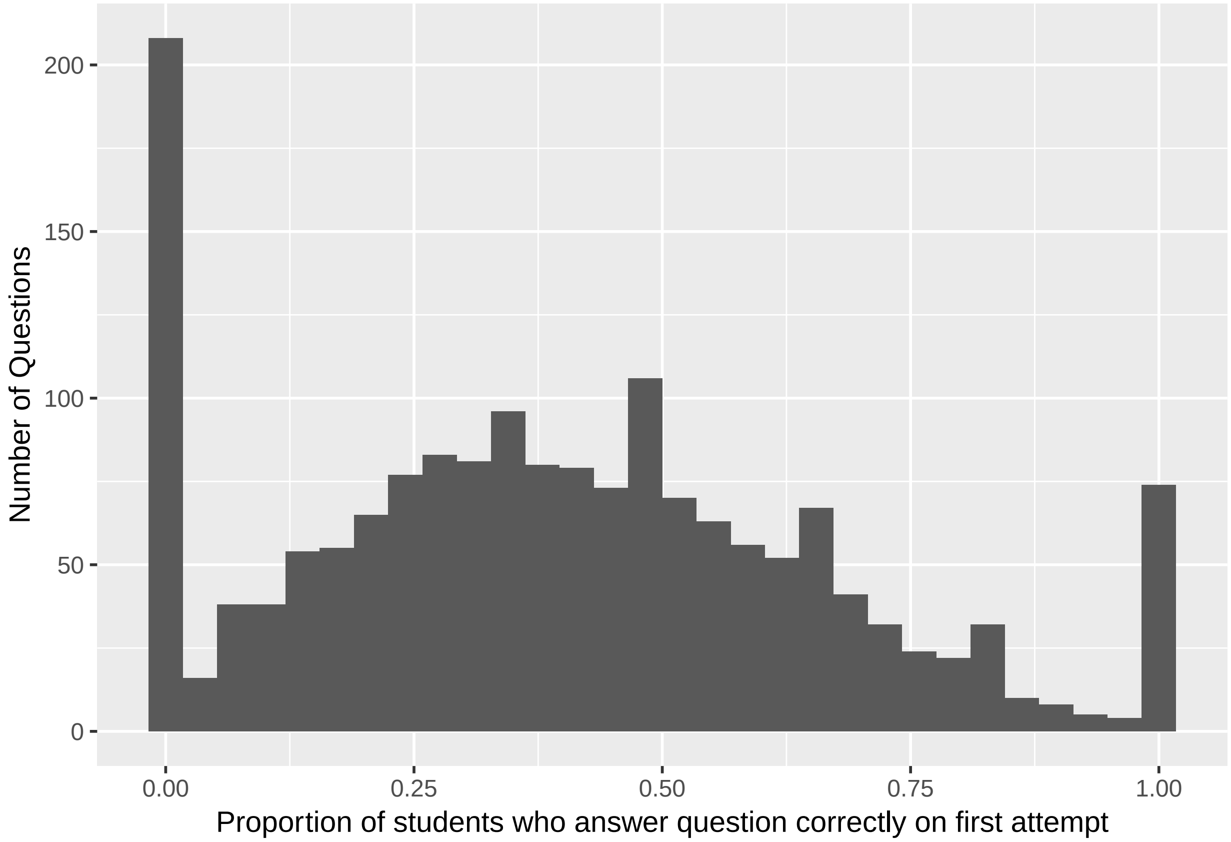


The Pearson correlation coefficient between the number of questions attempted and the proportion of questions answered correctly on the first attempt is ~0.101. Answering more questions does not appear to be a strong predictor of how likely a student is to answer correctly on the first attempt.

Q16: How difficult are the questions? Plot the question-level distribution of the proportion of students who get it right on the first attempt as a histogram. This quantitiy is called "item difficulty" (Tip: use the same approach as in Q13)

```
asm %>%
  group_by(itemid) %>%
  summarize(PropCorrectFirstAttempt = mean(correctonfirstattempt)) %>%
  ggplot(., aes(x = PropCorrectFirstAttempt)) +
  geom_histogram() +
  xlab('Proportion of students who answer question correctly on first attempt') +
  ylab('Number of Questions')
```

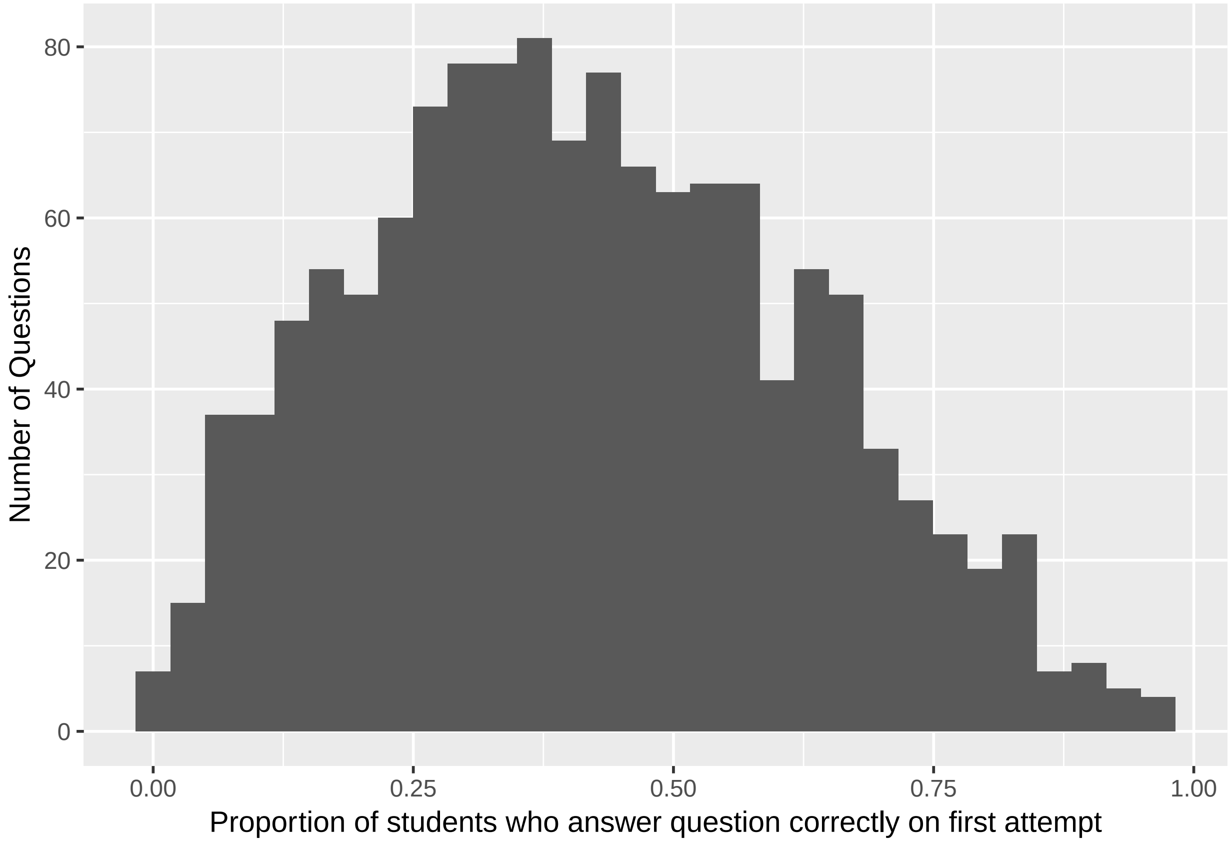
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Q17: Repeat Q16 but exclude questions that were attempted fewer than 10 times (note in the plot that this reduces the spikes at 0 and 1).

```
asm %>%
  group_by(itemid) %>%
  filter(n() > 9) %>%
  summarize(PropCorrectFirstAttempt = mean(correctonfirstattempt)) %>%
  ggplot(., aes(x = PropCorrectFirstAttempt)) +
  geom_histogram() +
  xlab('Proportion of students who answer question correctly on first attempt') +
  ylab('Number of Questions')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Exploring the Fractions Game Dataset

Print the first few lines of the fractions dataset:

```
head(fr)
```

```
##    ï..row SID             ip firstName X..Users Trials.per.user domain
## 1 1021160  75 170.185.142.19      1830        1               4  whole
## 2 1021177  75 170.185.142.19      1830        0               4  whole
## 3 1021196  75 170.185.142.19      1830        0               4  whole
## 4 1021215  75 170.185.142.19      1830        0               4  whole
## 5 1009151  54 170.185.142.19      5101        1             110  whole
## 6 1009164  54 170.185.142.19      5101        0             110  whole
##   experimentName       levelName Numberline currentLevelNo totalTrials
## 1    Condition-1 uclawhole1_0_10       0-10              1           4
## 2    Condition-1 uclawhole1_0_10       0-10              1           4
## 3    Condition-1 uclawhole1_0_10       0-10              1           4
## 4    Condition-1 uclawhole1_0_10       0-10              1           4
## 5    Condition-1 uclawhole1_0_10       0-10              1           4
## 6    Condition-1 uclawhole1_0_10       0-10              1           4
##   currentMode currentQuestion Opportunity.Count     Temp.Opp.
## 1         sub               8                 1 8user18300-10
## 2         sub               5                 1 5user18300-10
## 3         sub               1                 1 1user18300-10
## 4         sub               3                 1 3user18300-10
## 5         sub                 8                 1 8user51010-10
## 6         sub                 1                 1 1user51010-10
##        answerByUser answerDec Hit.Rate       hitType
## 1   (756.9 # 504.2)    7.8916        1 Perfect Hit!!
## 2  (522.65 # 509.1)    5.1355        1 Perfect Hit!!
## 3 (157.75 # 502.95)    0.8363        1 Perfect Hit!!
## 4   (287.2 # 500.5)    2.3675        0   Near Miss!!
## 5   (706.5 # 513.6)    7.3027        0   Near Miss!!
## 6   (192.9 # 467.8)    1.2485        1 Perfect Hit!!
##          currentTrialStartTime          currentTrialEndTime
## 1 Wed Sep 21 19:00:27 2011 UTC Wed Sep 21 19:00:38 2011 UTC
## 2 Wed Sep 21 19:00:43 2011 UTC Wed Sep 21 19:00:47 2011 UTC
## 3 Wed Sep 21 19:00:51 2011 UTC Wed Sep 21 19:00:57 2011 UTC
## 4 Wed Sep 21 19:01:02 2011 UTC Wed Sep 21 19:01:08 2011 UTC
## 5 Tue Sep 20 14:18:16 2011 UTC Tue Sep 20 14:18:24 2011 UTC
## 6 Tue Sep 20 14:18:29 2011 UTC Tue Sep 20 14:18:32 2011 UTC
##   currentAccuracy avgAccuracy curReactionTime   totalTime itemsPlayed
## 1           98.92       98.92          11.020 30.797 Secs           1
## 2           98.65       98.79           4.078 39.687 Secs           2
## 3           98.36       98.65           5.844 50.344 Secs           3
## 4           93.68       97.41           6.058 61.232 Secs           4
## 5           93.03       93.03           8.030   36.8 Secs           1
## 6           97.52       95.28           2.309 43.913 Secs           2
##   timeLimit   avgTime bestTime currentStarCount totalStarCount score
## 1        60 11.020000   11.020                1              1     0
## 2        60  7.549000    4.078                2              3     0
## 3        60  6.980667    4.078                3              6     0
## 4        60  6.750000    4.078                3              9     0
## 5        60  8.030000    8.030                0              0     0
## 6        60  5.169500    2.309                1              1     0
##   fireType sound        verInfo
## 1    CLICK    ON BSNL v2.1 UCLA
## 2    CLICK    ON BSNL v2.1 UCLA
## 3    CLICK    ON BSNL v2.1 UCLA
## 4    CLICK    ON BSNL v2.1 UCLA
## 5    CLICK    ON BSNL v2.1 UCLA
## 6    CLICK    ON BSNL v2.1 UCLA
```

There are more columns in this dataset and it requires watching the game video closely to understand what the variables could mean. On your own, go though each column as I did above and reason about what the variable could mean. It is not obvious that the column identifying users is *firstName*. Below are just a few questions.

Q18: Remove rows from the dataset where the *firstName* column is empty. How many unique students are there?

```
length(unique(fr[!is.na('firstName'), c('firstName')]))
```

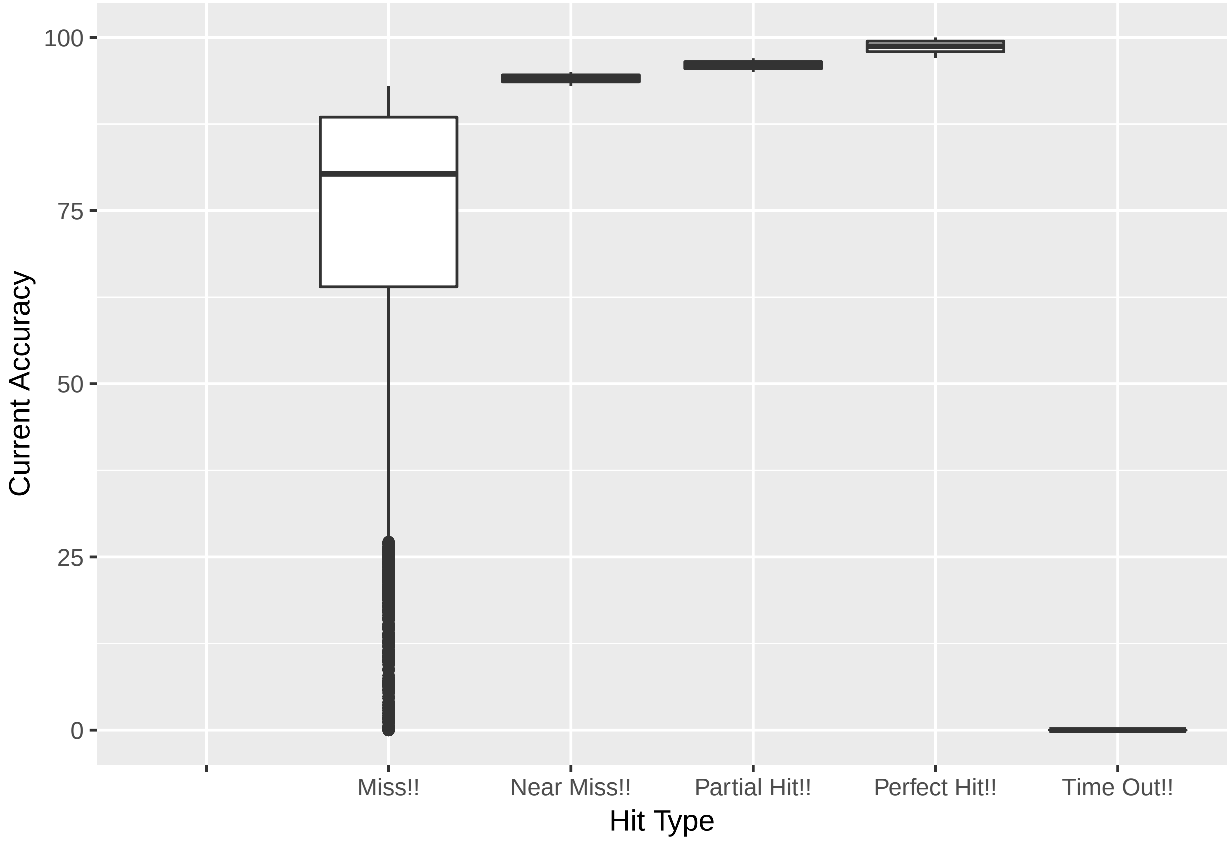
```
## [1] 118
```

There are 118 unique students in the dataset.

Q19: How does the game judge between a Miss, Near Miss, Partial Hit, and Perfect Hit? Make a boxplot of current accuracy against the hit type to investigate.

```
ggplot(fr, aes(x = hitType, y = currentAccuracy)) +
  geom_boxplot() +
  xlab('Hit Type') +
  ylab('Current Accuracy')
```

```
## Warning: Removed 5 rows containing non-finite values (stat_boxplot).
```



A ‘Time Out!’ corresponds to an accuracy of 0, while a ‘Miss!’ looks to be anything with accuracy less than about 90. Slightly above this threshold corresponds to a ‘Near Miss!’. Higher still corresponds to a ‘Partial Hit!’. At close to (but not necessarily exactly) 100% accuracy, this is a ‘Perfect Hit!’

Q20: Find the variable that best predicts the current accuracy out of the following: *Trials.per.user, currentLevelNo, curReactionTime, itemsPlayed, currentStarCount*?

```
cor(fr[, c('currentAccuracy', 'Trials.per.user', 'currentLevelNo', 'curReactionTime',
           'itemsPlayed', 'currentStarCount')], use = "pairwise.complete.obs")[2:6,
                                                                               c('currentAccuracy')]
```

```
##  Trials.per.user   currentLevelNo  curReactionTime      itemsPlayed
##       0.02229434      -0.13102657      -0.03753919      -0.12405685
## currentStarCount
##       0.36820004
```

Out of these variables ‘currentStarCount’ is most highly correlated with current accuracy and will better predict this variable than the other four variables.

Submit Homework

This is the end of the homework. Please **Knit a PDF report** that shows both the R code and R output and upload it on the EdX platform.