

Exploring Educational Datasets

INFO 5200 Learning Analytics: Week 2 Homework

[[ADD YOUR NAME, CORNELL ID]]

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 Assisstmnts 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 import and export any object in R. This can be a scalar, 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 Assisstmnts dataset (*info5200.2.assisstmnts.rds*) into R and call it `asm`.
- Load the fraction game dataset (*info5200.2.gamedata.csv*) into R and call it `fr`.

```
asm = readRDS("info5200.2.assisstmnts.rds")
fr = read.csv("info5200.2.gamedata.csv")
```

Exploring the Assisstmnts 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()`.

```
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?

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

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

Q2: How many unique questions are there?

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

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

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

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

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

Q4: What is the rate of asking for hints?

```
mean(asm$hints > 0)
```

```
## [1] 0.326534
```

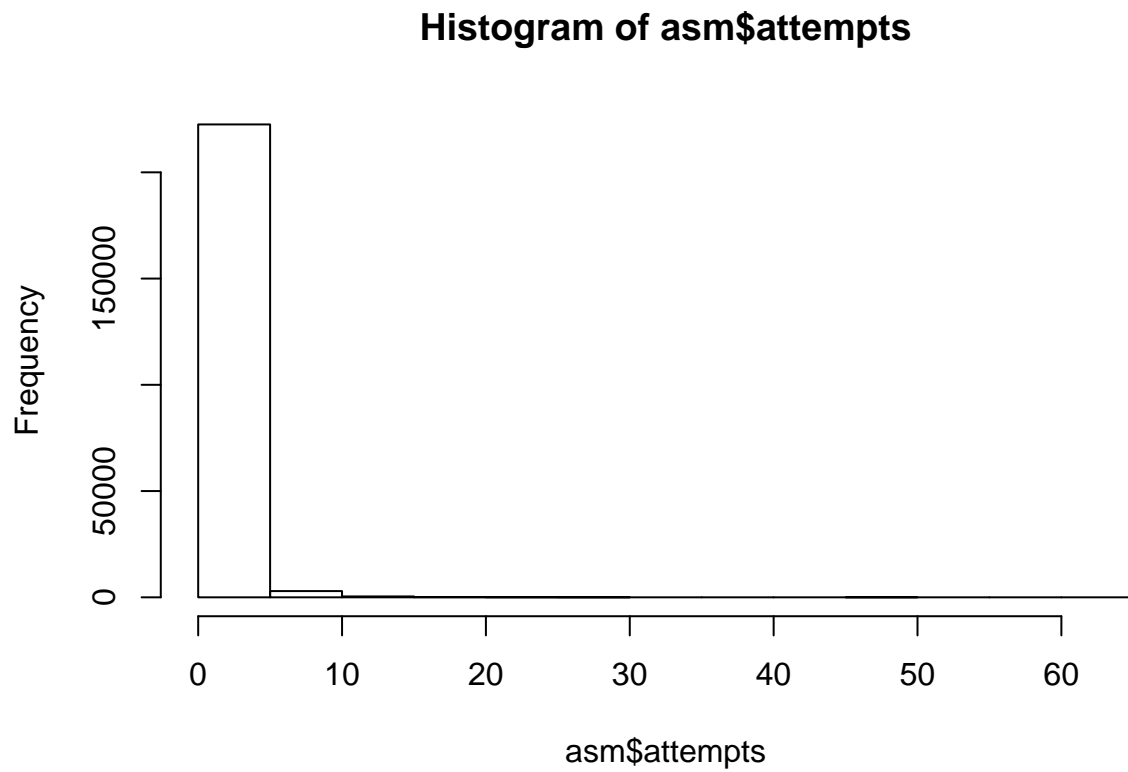
Q5: How long do students spend on a question on average?

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

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

Q6: Plot the distribution of attempts as a histogram:

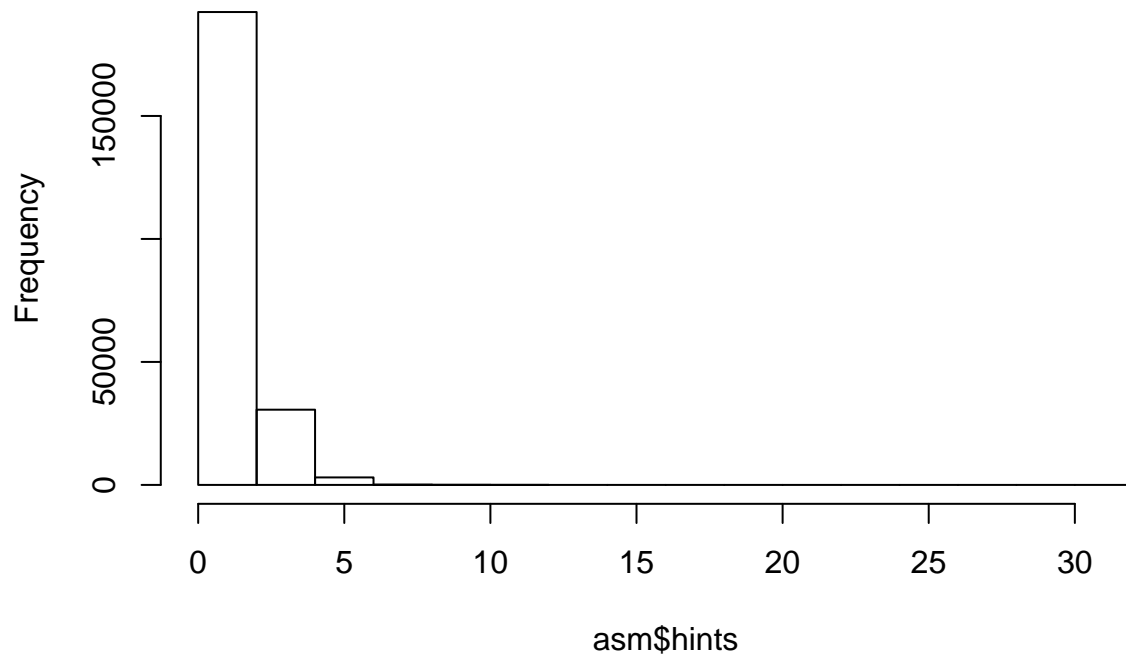
```
hist(asm$attempts)
```



Q7: Plot the distribution of hints as a histogram:

```
hist(asm$hints)
```

Histogram of asm\$hints



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

```
cor(asm$seconds, asm$attempts)
```

```
## [1] 0.4345258
```

```
cor(asm$attempts, asm$hints)
```

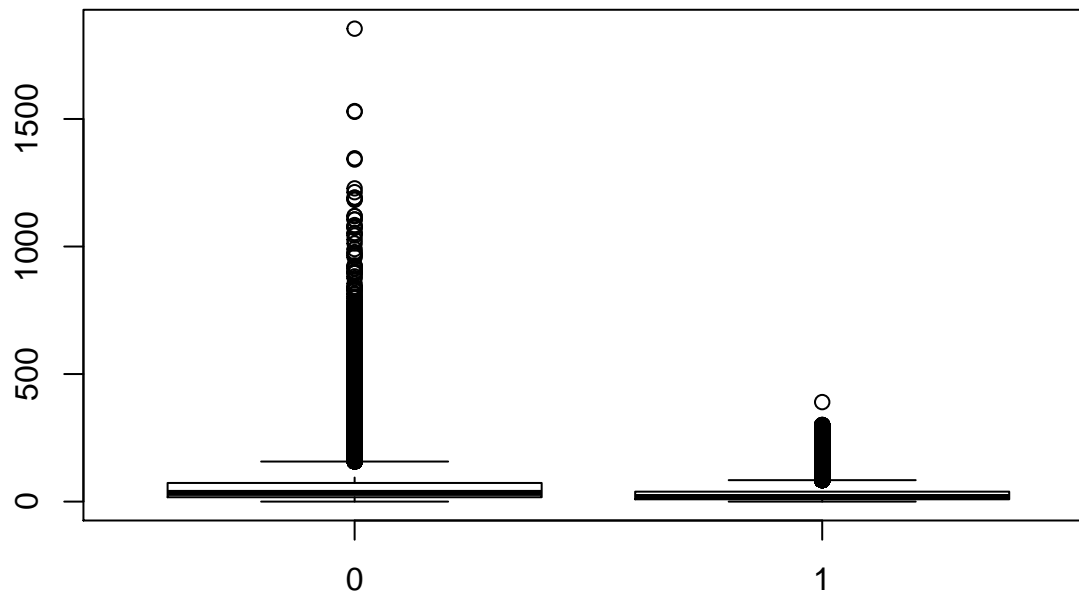
```
## [1] 0.1275844
```

```
cor(asm$seconds, asm$hints)
```

```
## [1] 0.1709763
```

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:

```
boxplot(asm$seconds ~ asm$correctonfirstattempt)
```



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):

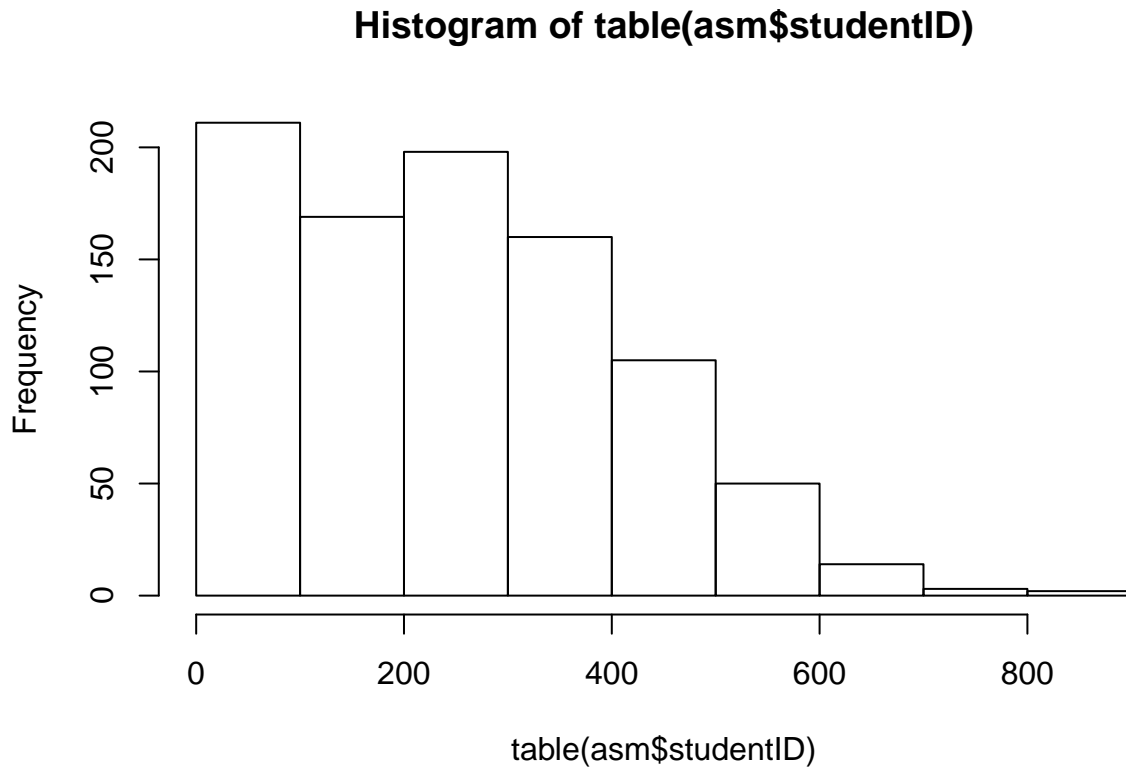
```
table(asm$hints, asm$correctonfirstattempt)
```

```
##
##          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:

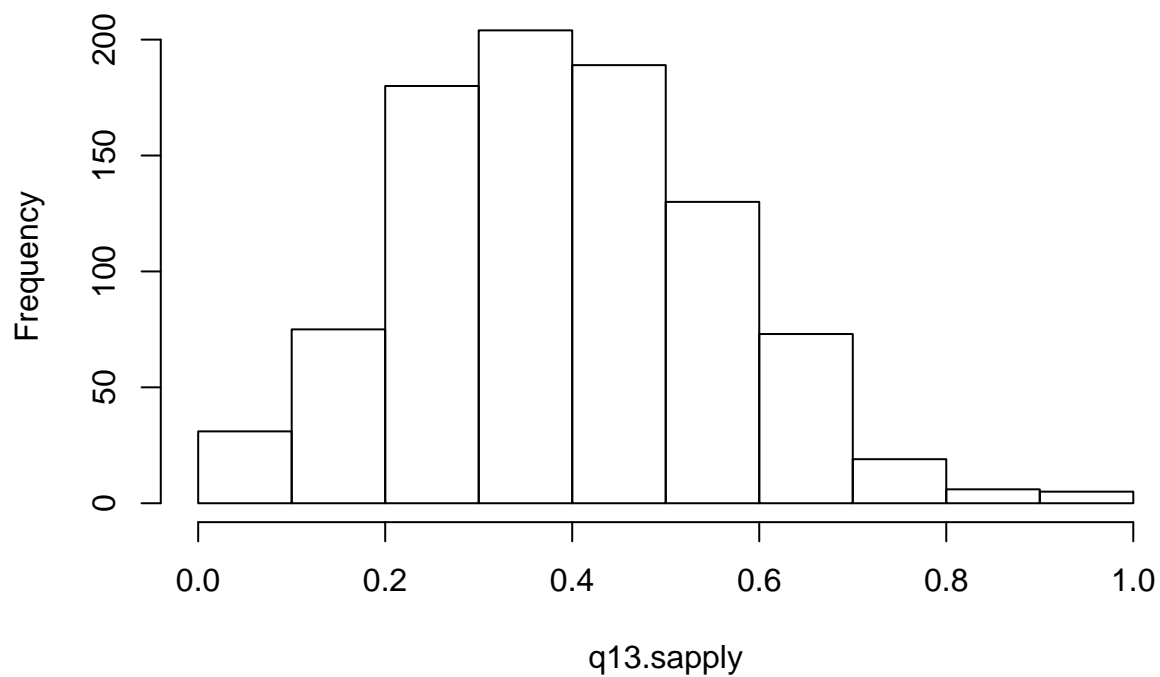
```
hist(table(asm$studentID))
```



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):

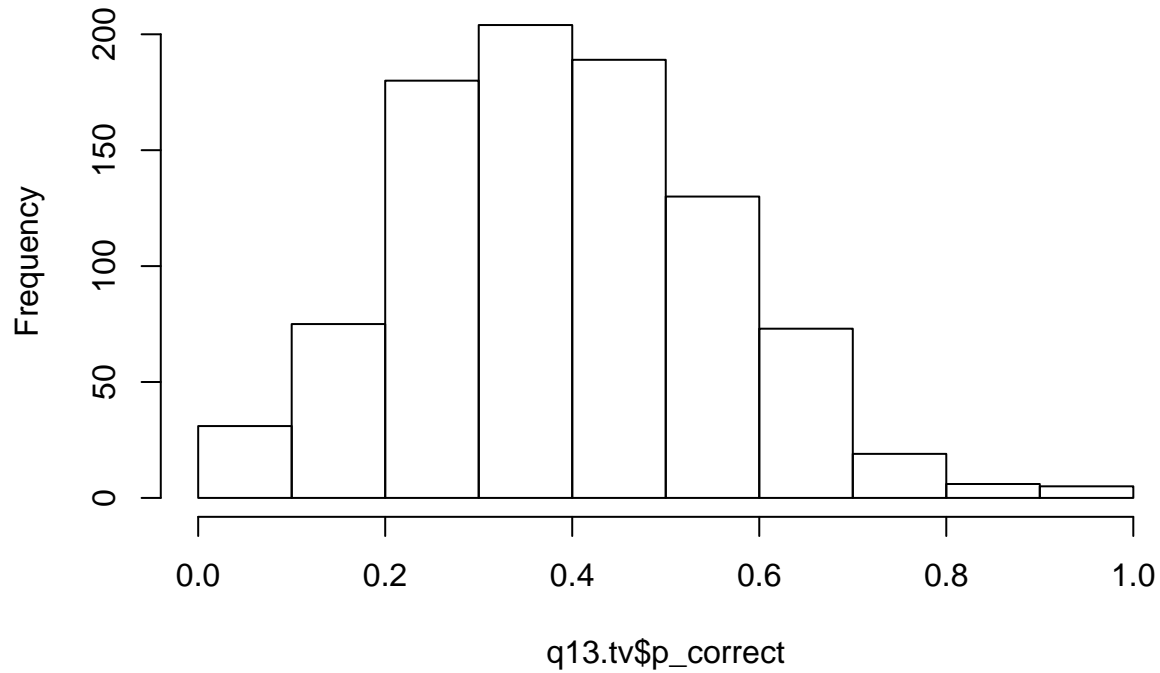
```
# Base R with sapply: loop through IDs and get proportion of
# correct on first attempt for all questions for that ID
q13.sapply = sapply(
  X = unique(asm$studentID),
  FUN = function (id) mean(asm$correctonfirstattempt[asm$studentID == id])
)
hist(q13.sapply)
```

Histogram of q13.sapply



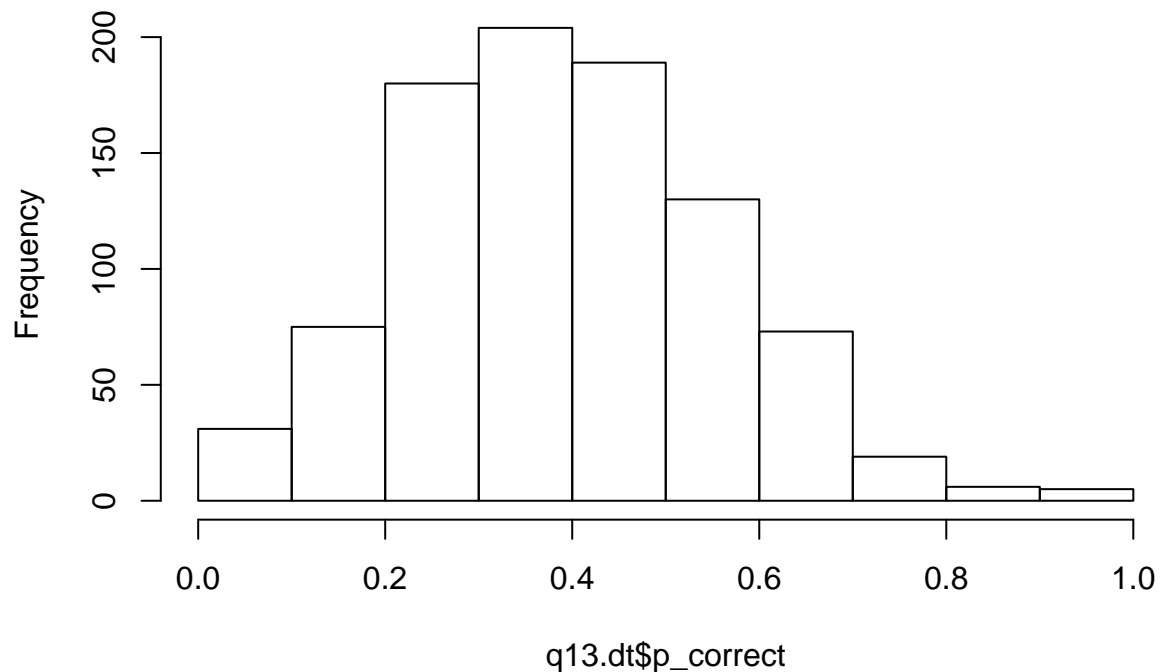
```
# Using tidyverse: group by student and then summarise the proportion
library(tidyverse, quietly = T)
q13.tv = asm %>%
  group_by(studentID) %>%
  summarise(p_correct = mean(correctonfirstattempt))
hist(q13.tv$p_correct)
```

Histogram of q13.tv\$p_correct



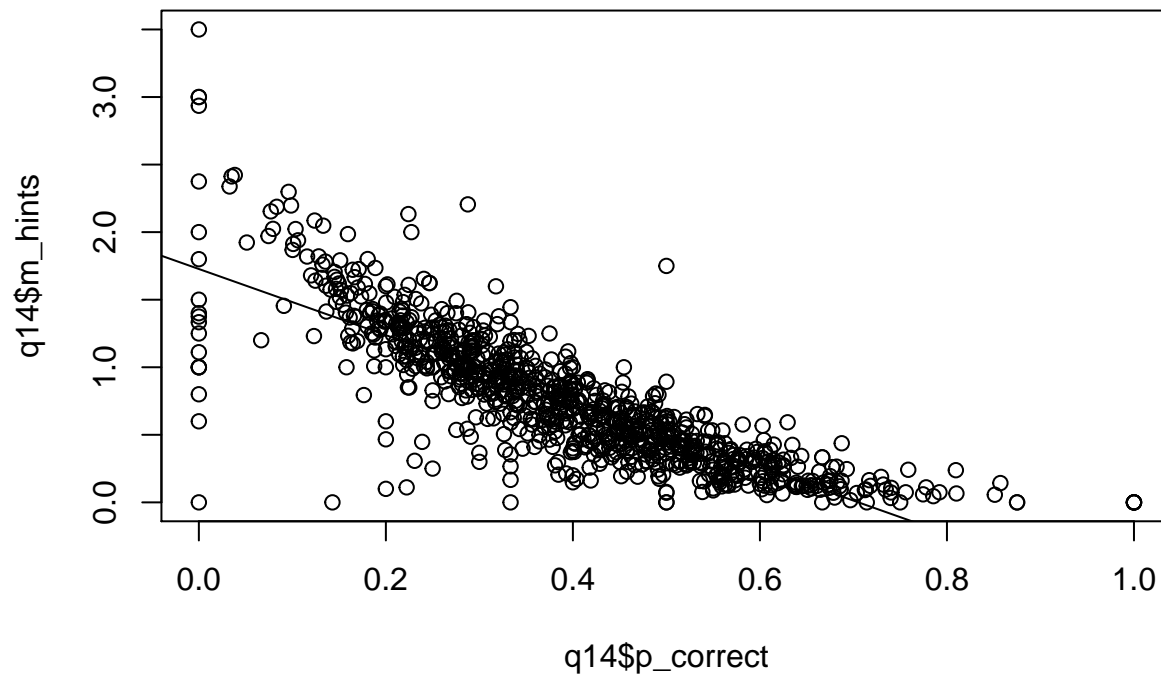
```
# Using data.table: need to make convert dataset from data.frame to  
# data.table object first, then apply `.` and `by` syntax  
library(data.table, quietly = T)  
q13.dt = data.table(asm)[,.(p_correct = mean(correctonfirstattempt)), by = studentID]  
hist(q13.dt$p_correct)
```

Histogram of q13.dt\$p_correct



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.

```
# Using the tidyverse package
q14 = asm %>%
  group_by(studentID) %>%
  summarise(
    p_correct = mean(correctonfirstattempt),
    m_hints = mean(hints)
  )
plot(q14$p_correct, q14$m_hints)
abline(lm(q14$m_hints ~ q14$p_correct))
```

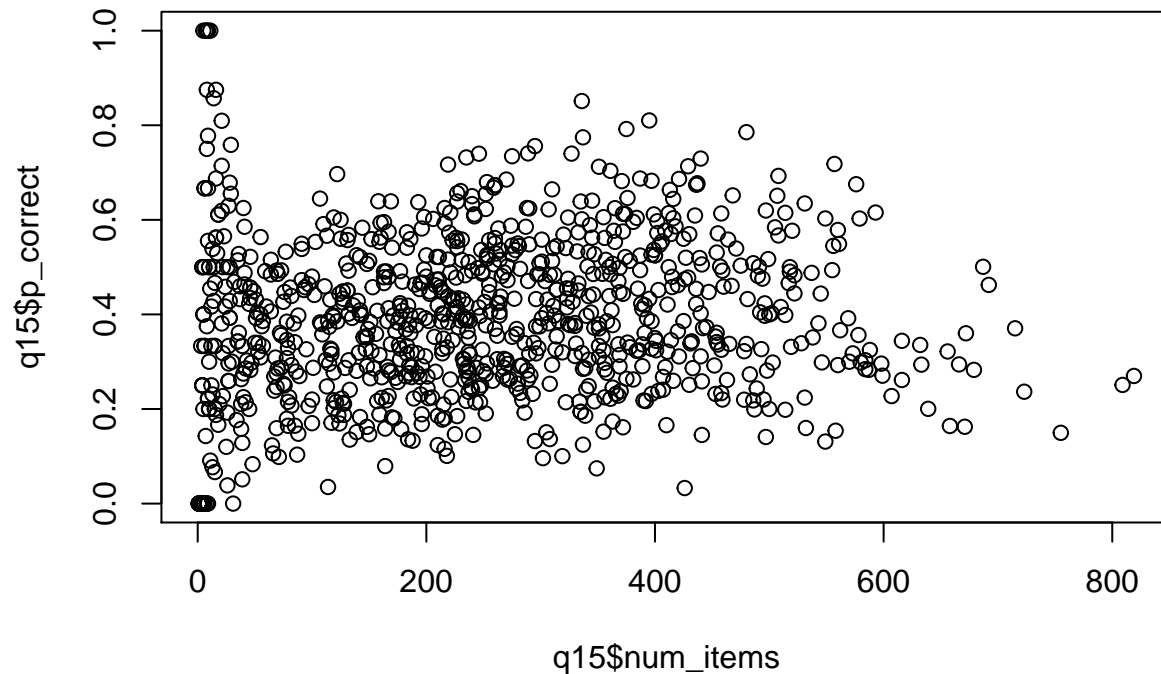


Q15: Are students who attempt more questions (i.e. get more practice) more likely to answer correctly on the first attempt? Provide a correlation and scatterplot.

```
q15 = asm %>%
  group_by(studentID) %>%
  summarise(
    p_correct = mean(correctonfirstattempt),
    num_items = n() # n() returns a count of rows
  )
cor(q15$num_items, q15$p_correct)
```

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

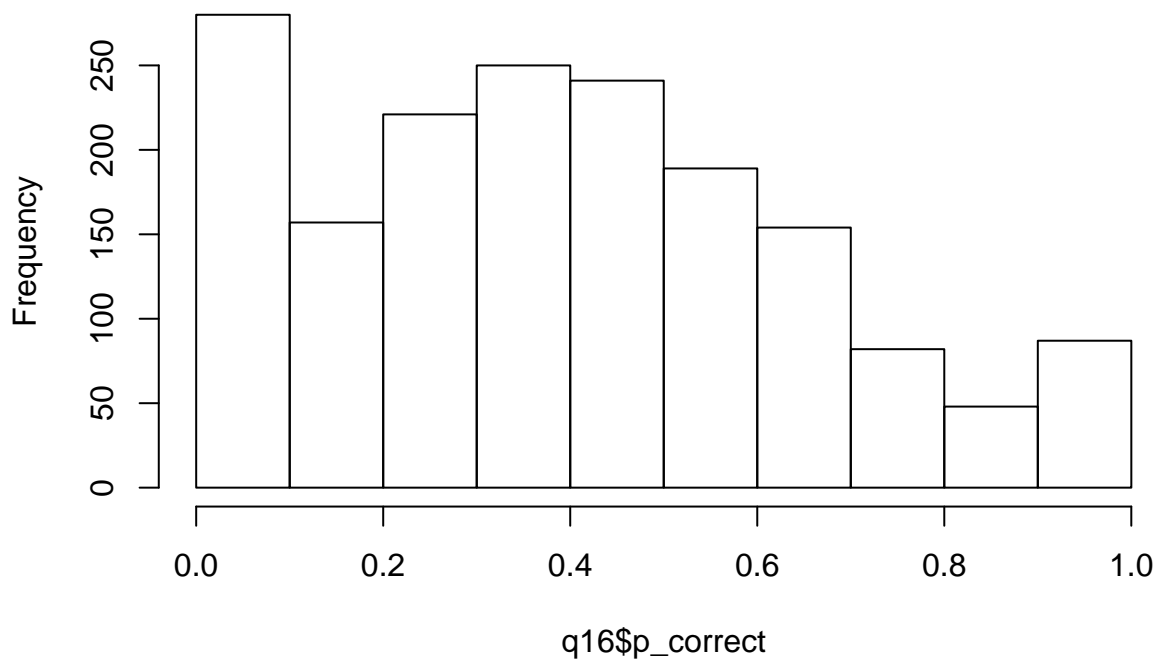
```
plot(q15$num_items, q15$p_correct)
```



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 quantity is called “item difficulty” (Tip: use the same approach as in Q13)

```
q16 = asm %>%
  group_by(itemid) %>%
  summarise(p_correct = mean(correctonfirstattempt))
hist(q16$p_correct)
```

Histogram of q16\$p_correct

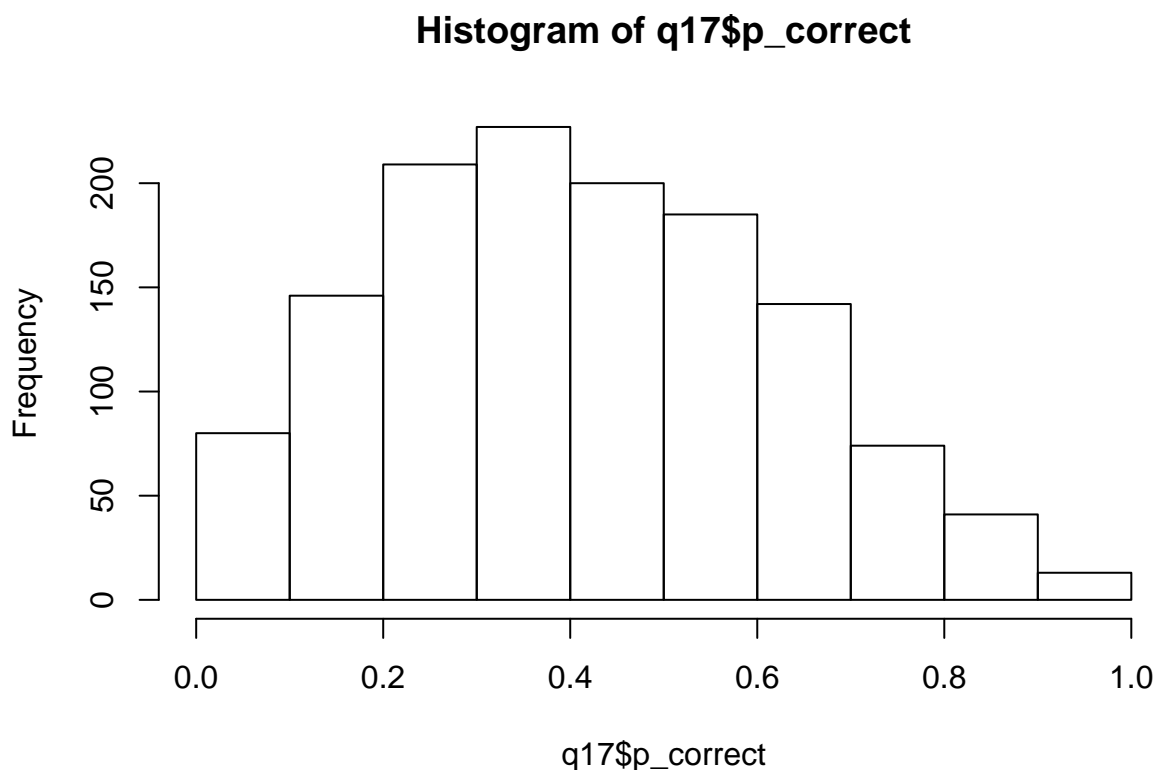


Q17: Repeat Q16 but exclude questions that were attempted fewer than 10 times (note in the plot that this reduces the spikes at 0 and 1).

```
q17 = asm %>%
  group_by(itemid) %>%
  summarise(
    p_correct = mean(correctonfirstattempt),
    n = n()
  ) %>%
  filter(n >= 10)
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

```
hist(q17$p_correct)
```



Exploring the Fractions Game Dataset

```
head(fr)
```

```
##      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 through 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.

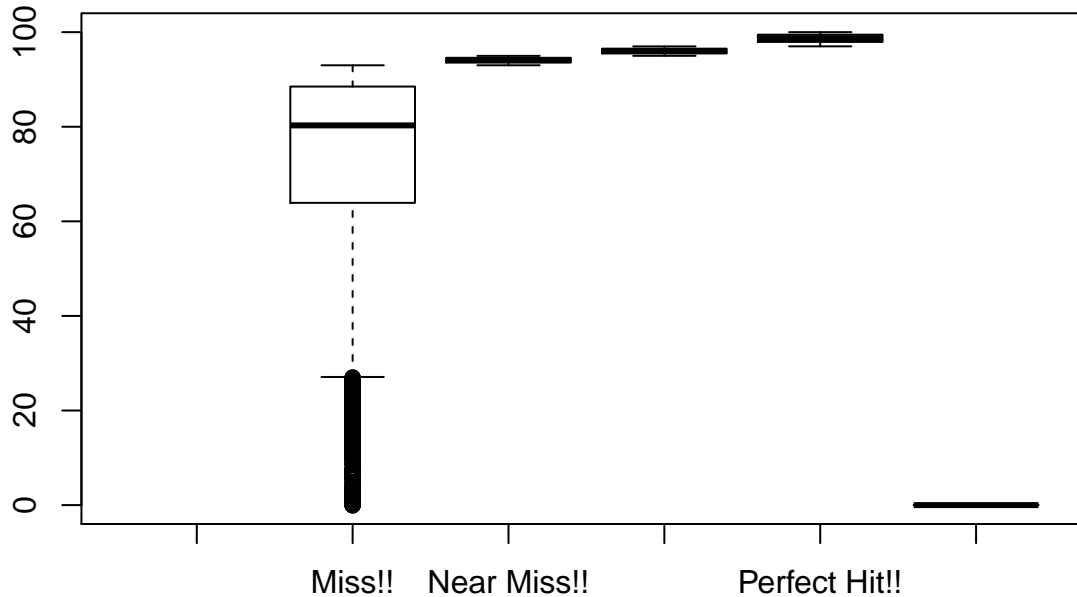
Q18: Remove rows from the dataset where the *firstName* column is empty. How many unique students are there?

```
fr = fr %>% filter(firstName != "")
n_distinct(fr$firstName) # like length(unique()), requires tidyverse
```

```
## [1] 117
```

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.

```
boxplot(fr$currentAccuracy ~ fr$hitType)
```



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

```
options(digits = 2) # round to two sig. digits
fr %>%
  select(currentAccuracy, Trials.per.user, currentLevelNo,
         curReactionTime, itemsPlayed, currentStarCount) %>%
  cor # cor() computes the pair-wise Pearson correlations
```

```
##           currentAccuracy Trials.per.user currentLevelNo
## currentAccuracy           1.000           0.022        -0.132
## Trials.per.user           0.022           1.000           0.185
## currentLevelNo          -0.132           0.185           1.000
## curReactionTime          -0.043          -0.155          -0.069
## itemsPlayed             -0.126           0.197           0.979
## currentStarCount         0.368           0.017          -0.040
##           curReactionTime itemsPlayed currentStarCount
## currentAccuracy          -0.043          -0.126           0.368
## Trials.per.user          -0.155           0.197           0.017
## currentLevelNo          -0.069           0.979          -0.040
## curReactionTime           1.000          -0.068          -0.021
## itemsPlayed              -0.068           1.000          -0.022
## currentStarCount         -0.021          -0.022           1.000
```

```
# currentStarCount has the highest correlation
```

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.