Exploring Educational Datasets

INFO 4100 Learning Analytics Homework

[[ADD YOUR NAME, CORNELL ID]]

In this homework, you will conduct an exploratory analysis with a public datasets obtained from PSLC DataShop. The dataset provides question-level data of students practicing math problems in academic year 2004-05 using the Assisstments platform. On this platform, students can attempt a problem many times to get it right and they can ask for several hints (one at a time) on a problem until the final hint tells them what the answer is.

Learning Objectives:

1. Identify a dataset file format and use the appropriate function to load it.
2. Explore fundamental properties of a dataset using basic functions in R
3. Compute and visualize relationships between variables using correlations, histograms, boxplots, and scatterplots
4. Calculate and visualize sutdent- and question-level quantities and relationships

Part 1: Loading the Dataset

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. Saving a dataset as an RDS file is much more efficient (smaller file size) than saving it as a CSV. Use the Help panel in RStudio to see examples of how to use any of these functions.

Question 1: Load the Assistments dataset (*info4100.data.assisstments.rds*) into R and call it `asm`.

```
#######################################
####### BEGIN INPUT: Question 1 #######
#######################################
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.4.4     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.0
## ✔ purrr     1.0.2
## ── Conflicts ──────────────────────────────────────── tidyverse_conflicts()
──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
asm = readRDS("info4100.data.assisstments.rds")
```

```
########################################
########################################
```

Part 2: Exploring the Dataset

It is hard to overstate the importance of understanding the data you are working with. Ideally, you understand the data-generating process, that is, how the data was collected or logged by the system. To start off, you should check what is in the dataset.

Question 2: To look at the first few rows of `asm`, use the `head()` function. (Note that because this is a .Rmd R-markdown file the output will appear inline below the R block.)

```
########################################
####### BEGIN INPUT: Question 2 #######
########################################
```

```
head(asm)
```

	studentID	itemid	correctonfirstattempt	attempts	hints	seconds	full_start_time
	<int>	<int>	<int>	<int>	<int>	<int>	<fct>
1	136	90	1	1	0	58	01-OCT-04 07.44.43.000000 AM

2	136	91	0	1	3	91	10-DEC-04 09.27.20.000000 AM
3	136	92	1	1	0	11	10-DEC-04 09.28.51.000000 AM
4	136	93	1	1	0	10	10-DEC-04 09.29.02.000000 AM
5	136	94	0	2	0	43	10-DEC-04 09.29.12.000000 AM
6	136	95	1	1	0	13	15-OCT-04 07.44.14.000000 AM

6 rows | 1-8 of 13 columns

```
########################################
########################################
```

Look at the first few rows and use the small arrow on the right side of the output to see additional columns that may be cut off due to space. 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 attempts used
- *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 roll-up. The data is at the student-question level, which means that there is one row for each question a student answered that summarizes the interaction with the question (performance indicators and time spent).

Now you will answer a series of questions about this dataset. It is the kind of questions you would provide answers to if you were to write a report about the dataset. Be sure to show your R code and the final answer to the question inside of the input area.

Question 3: How many unique students are in the dataset? Tip: Use the `unique()` function and the `length()` function.

```
#######################################
####### BEGIN INPUT: Question 3 #######
#######################################

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

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

```
#######################################
#######################################
```

Question 4: How many unique questions are in the dataset?

```
#######################################
####### BEGIN INPUT: Question 4 #######
#######################################

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

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

```
#######################################
#######################################
```

Question 5: What proportion of first attempts do students get right?

```
#######################################
####### BEGIN INPUT: Question 5 #######
#######################################

mean(asm$correctonfirstattempt)
```

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

```
#######################################
#######################################
```

Question 6: What is the proportion of questions for which students request one or more hints?

```
#######################################
####### BEGIN INPUT: Question 6 #######
#######################################

#solution 1:
mean(asm$hints > 0)
```

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

```
#solution 2:
q_level = asm %>%
  group_by(itemid) %>%
  filter(min(hints) > 0)
nrow(q_level)/length(unique(asm$itemid))
```

```
## [1] 0.144529
```

```
#######################################
#######################################
```

Question 7: How long do students spend on a question on average? Tip: Use the `mean()` function.

```
#######################################
####### BEGIN INPUT: Question 7 #######
#######################################

mean(asm$seconds)
```

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

```
#######################################
#######################################
```

Question 8: What is the frequency distribution of different numbers of hints? That is, on how many questions did students need no hint (0), one hint, two hints, etc.? Tip: Use the `table()` function. You are not asked to visualize it.

```
#######################################
####### BEGIN INPUT: Question 8 #######
#######################################

table(asm$hints)
```

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

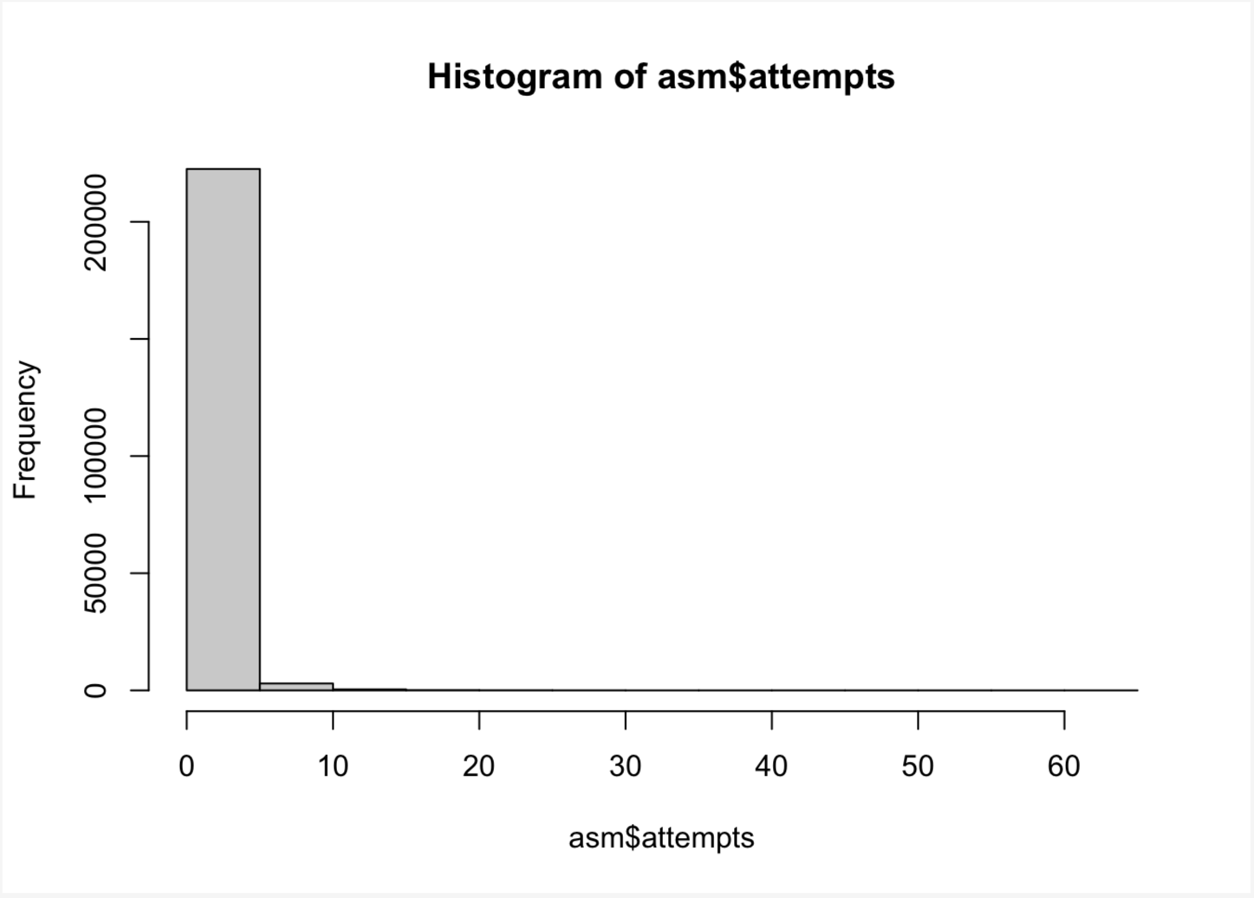
```
#######################################
#######################################
```

Part 3: Visualizing Distributions of Variables

Question 9: How many attempts do students make on questions? To answer, plot the distribution of attempts using a histogram. Tip: Use the `hist()` function.

```
########################################
####### BEGIN INPUT: Question 9 #######
########################################
```

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



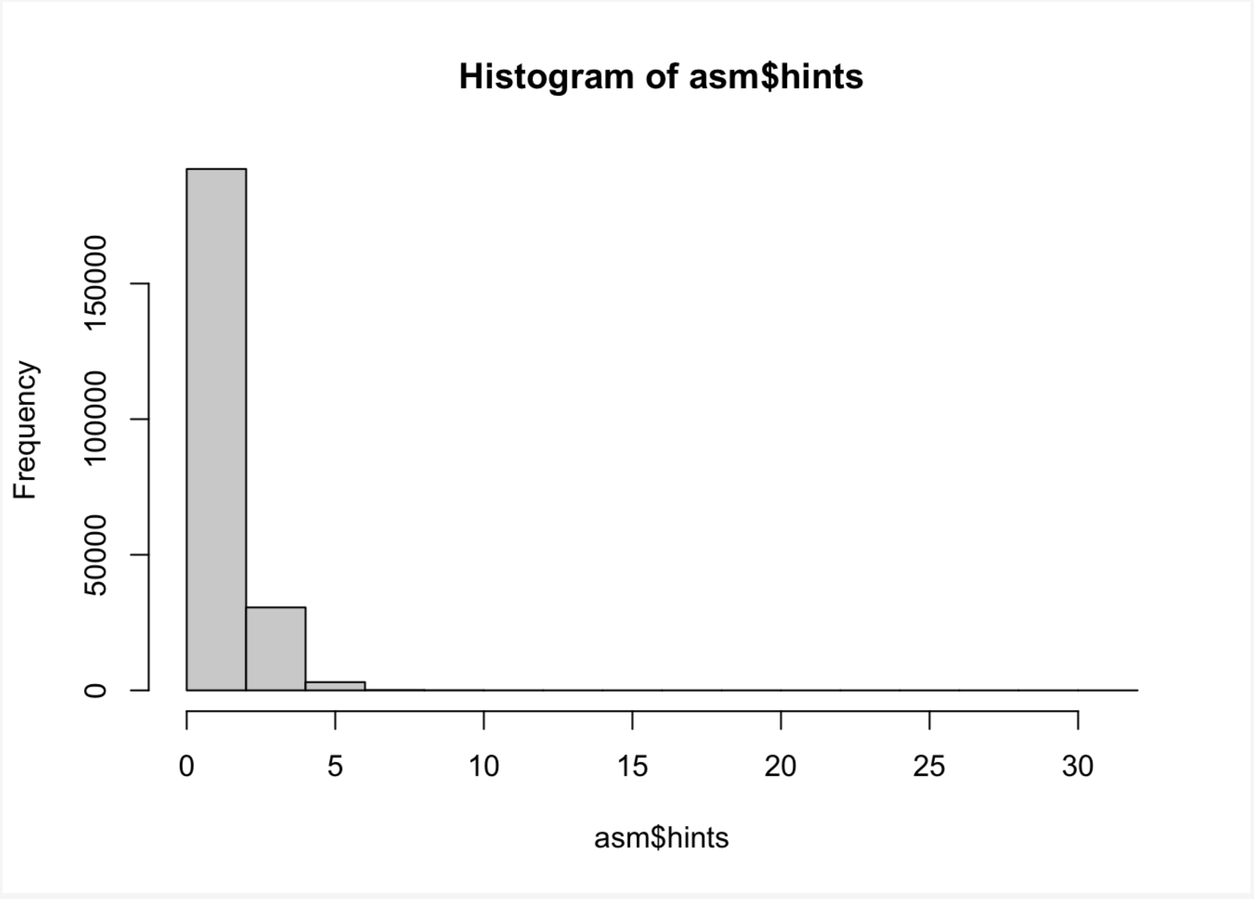
```
########################################
########################################
```

Question 10: How many hints do students ask for? To answer, plot the distribution of hints.

```
########################################
####### BEGIN INPUT: Question 10 ######
```

```
#######################################
```

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

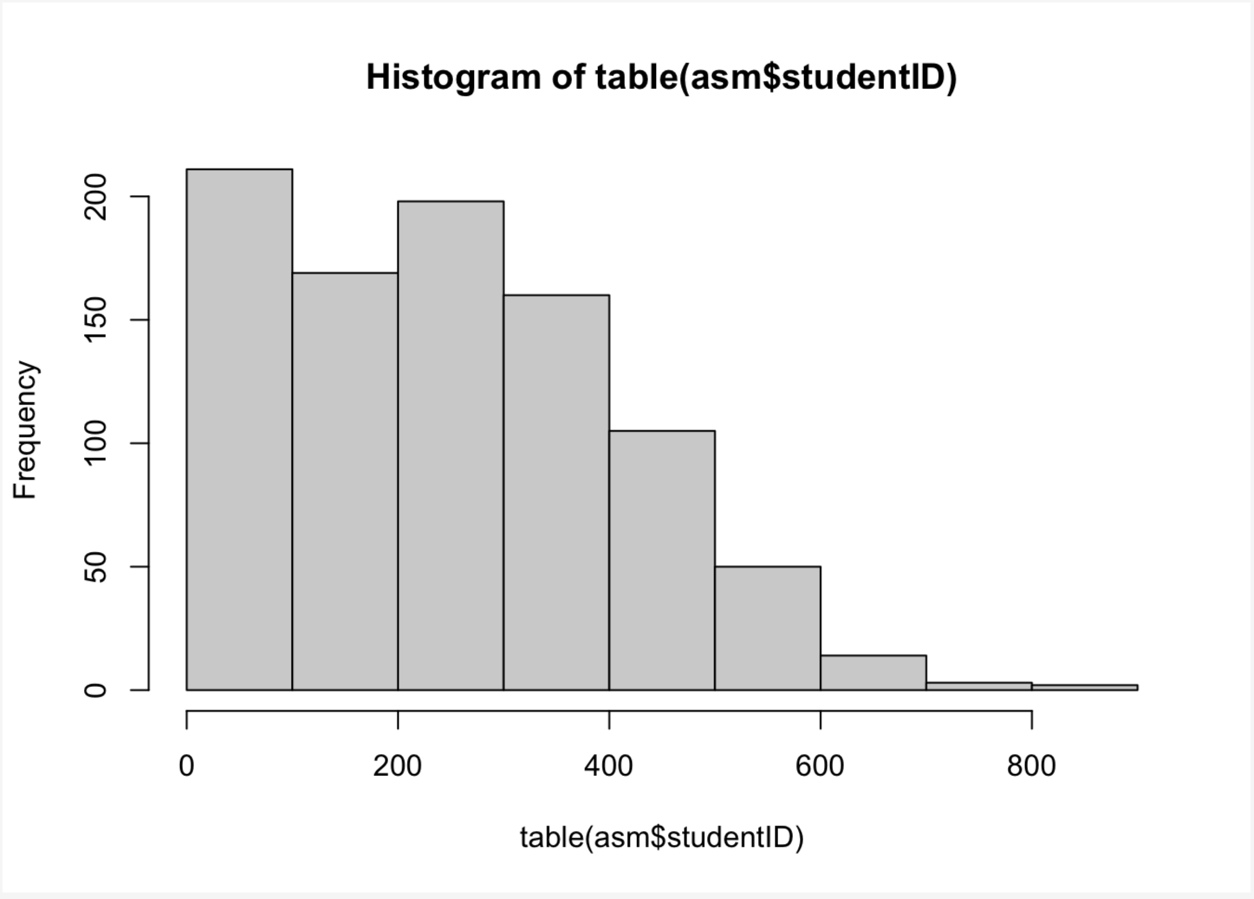


```
#######################################
#######################################
```

Question 11: How engaged are students on the platform? To answer, plot the distribution of how many questions each student answered. Tip: This requires an intermediate step to calculate the frequency distribution of question answering before plotting a histogram.

```
#######################################
####### BEGIN INPUT: Question 11 ######
#######################################
```

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



```
#######################################
#######################################
```

Part 4: Exploring Relationships Between Variables

Question 12: How is time spent, number of attempts, and getting hints related to one another? To answer, compute the three pair-wise correlations between time spent, number of attempts, and number of hints? Tip: Use the `cor()` function.

```
#######################################
####### BEGIN INPUT: Question 12 ######
#######################################
```

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

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

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

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

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

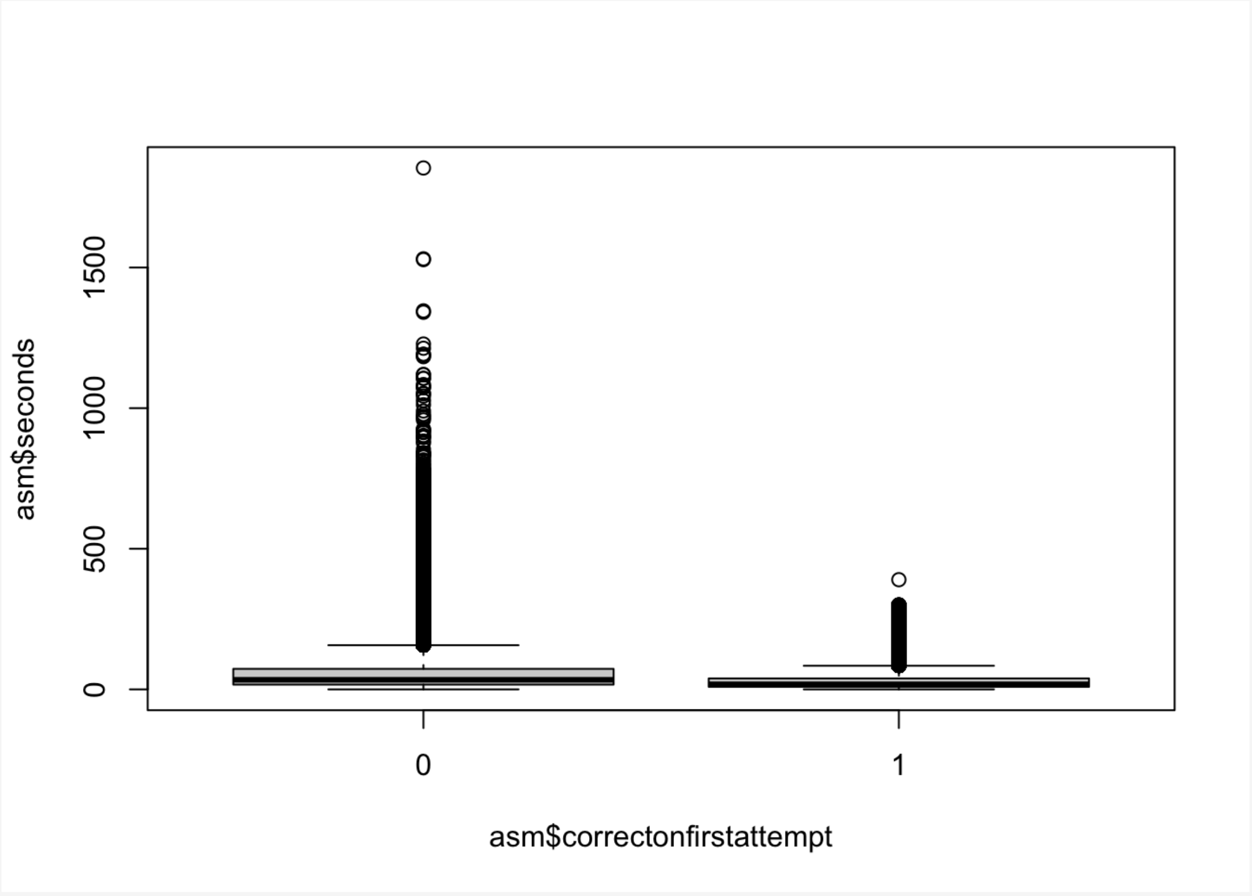
```
## [1] 0.1709763
```

```
########################################
########################################
```

Question 13: Do students spend more or less time answering questions that they get correct on the first attempt versus not? To answer, compare the distributions of time spent for questions that students got right on the first attempt versus those they didn't using a boxplot. Tip: Use the `boxplot()` function; look up the syntax in the help panel.

```
########################################
####### BEGIN INPUT: Question 13 ######
########################################
```

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



```
#######################################
#######################################
```

Question 14: How is asking for hints related to getting the right answer on the first attempt? To answer, cross-tabulate the frequency distribution of hints against getting it right on the first attempt. Tip: Use the `table()` function with two input variables. You should see in the output that 10 (6+2+2) students asked for hints before making an answer attempt and then got it right on their first attempt.

```
#######################################
####### BEGIN INPUT: Question 14 ######
#######################################
```

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

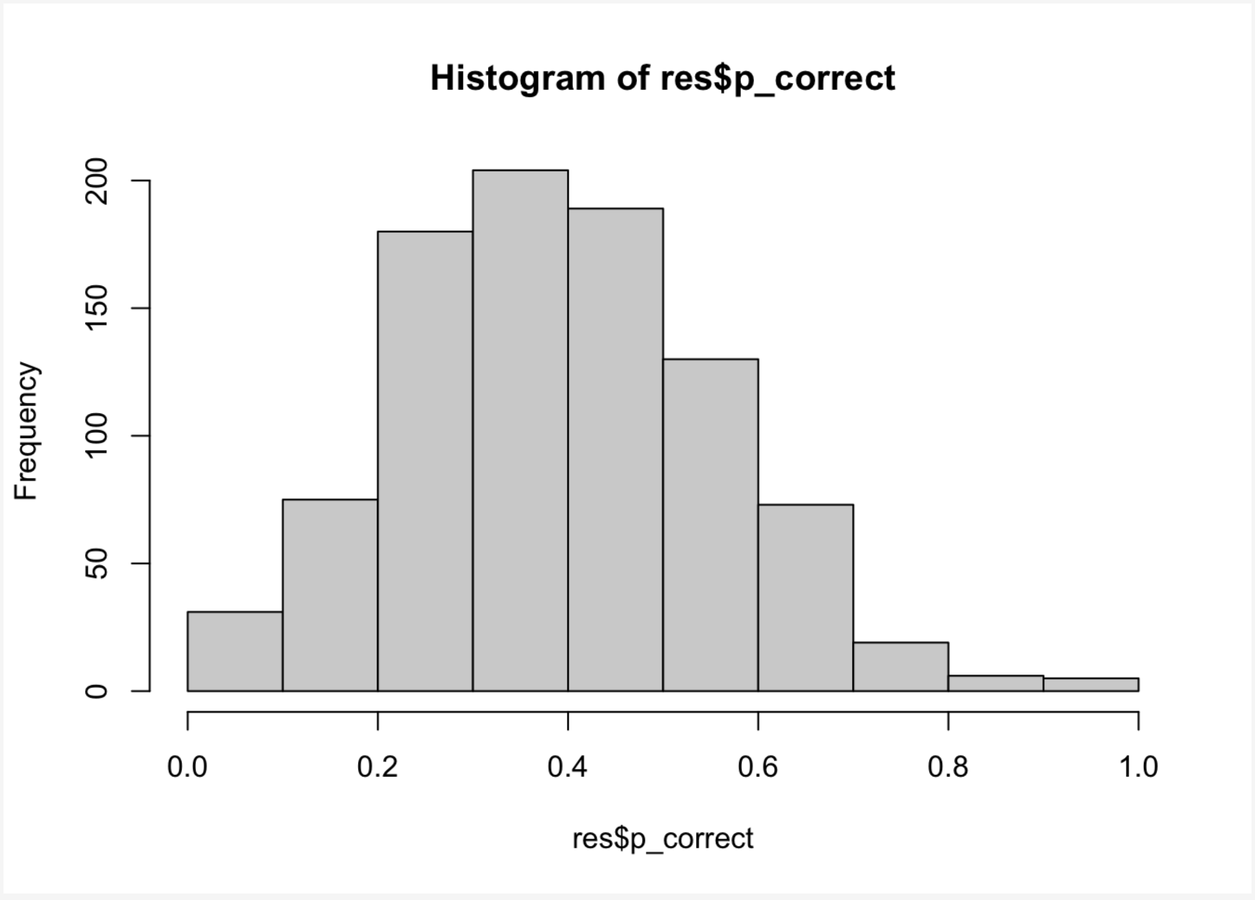
```
##     
##          0     1
##   0  60765 91505
##   1  25835     6
##   2  14137     2
##   3  23666     2
##   4   6916     0
##   5   2088     0
##   6    953     0
##   7     74     0
##   8     57     0
##   9     36     0
##   10    17     0
##   11     5     0
##   12    16     0
##   13     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
```

```
#######################################
#######################################
```

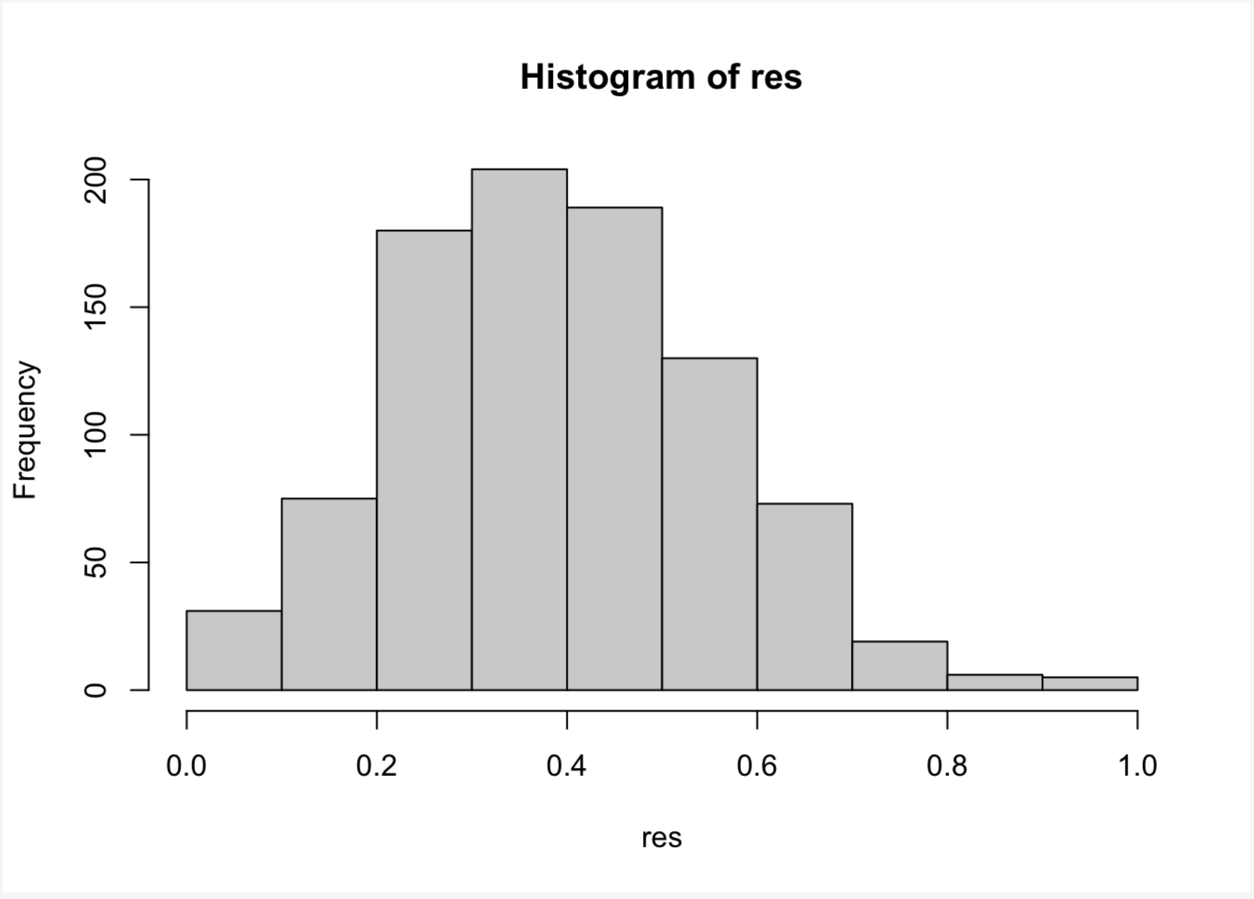
Part 5: Students and Questions as the Unit of Analysis

Question 15: How does the average student perform? To answer, plot the student-level distribution (i.e. 1 value per student) of answering correctly on the first attempt. The plot should be a histogram with the proportion answering correctly on the x-axis and the corresponding number of students on the y-axis. Tip: You first need to compute the proportion of questions that each student got right on their first attempt; there are several ways to do this. You can load the `tidyverse` package and use `group_by` and `summarise`; or you can use `sapply()`; or load the `data.table` package; whatever you do, don't use a *for* loop unless you really cannot solve it otherwise.

```
######################################
####### BEGIN INPUT: Question 15 ######
######################################

# Using tidyverse: group by student and then summarise the proportion
library(tidyverse)

res = asm %>%
    group_by(studentID) %>%
    summarise(p_correct = mean(correctonfirstattempt))
hist(res$p_correct)
```



```
# Base R with sapply: for each ID, get proportion of
#  correct on first attempt for all questions
res = sapply(
  X = unique(asm$studentID),
  FUN = function (id) mean(asm$correctonfirstattempt[asm$studentID == id])
)
hist(res)
```

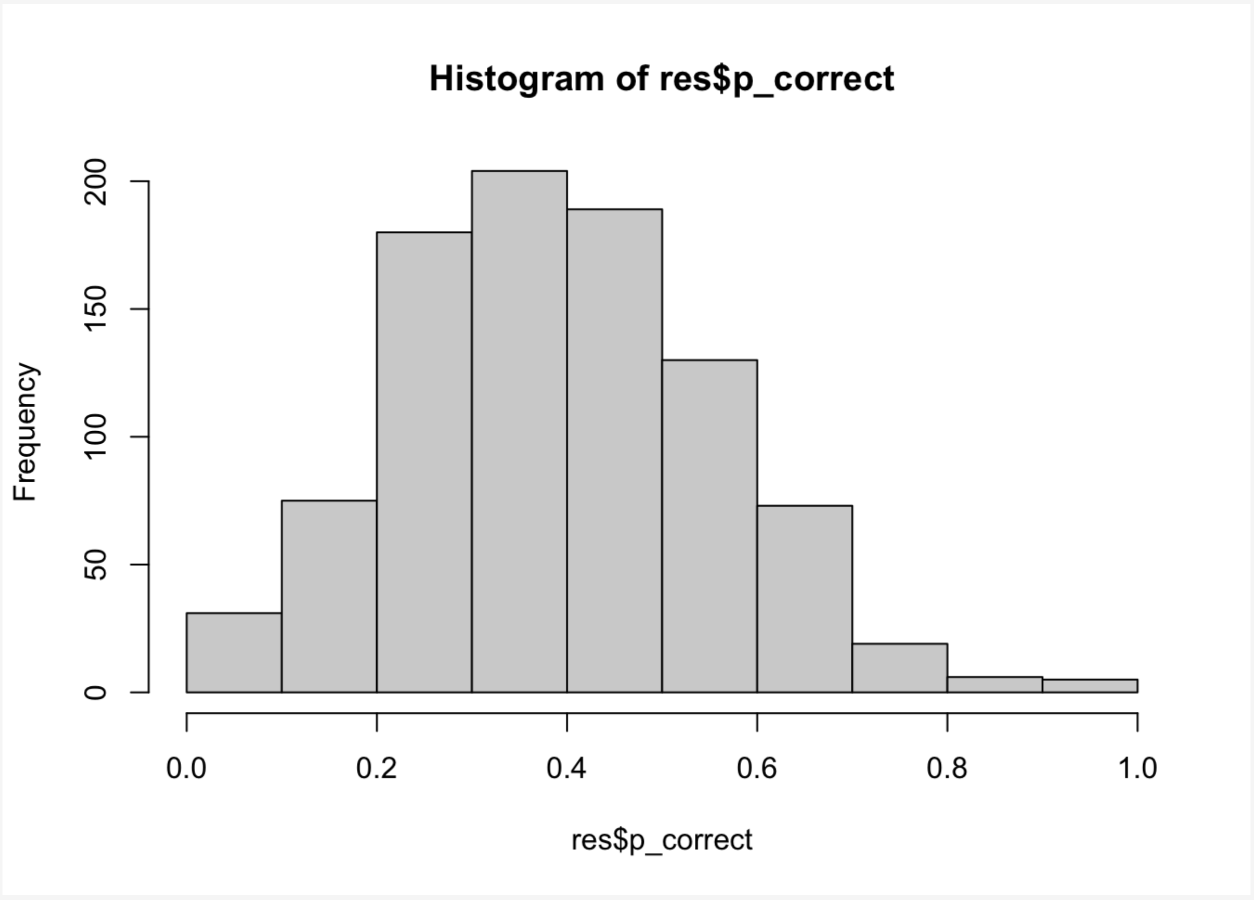


```
# Using data.table: need to convert dataset from a data.frame to
#  a data.table object first, then apply `.()` and `by` syntax
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

## The following object is masked from 'package:purrr':
##
##     transpose
```

```
res = data.table(asm)[,.(p_correct = mean(correctonfirstattempt)), by = studentID]
```

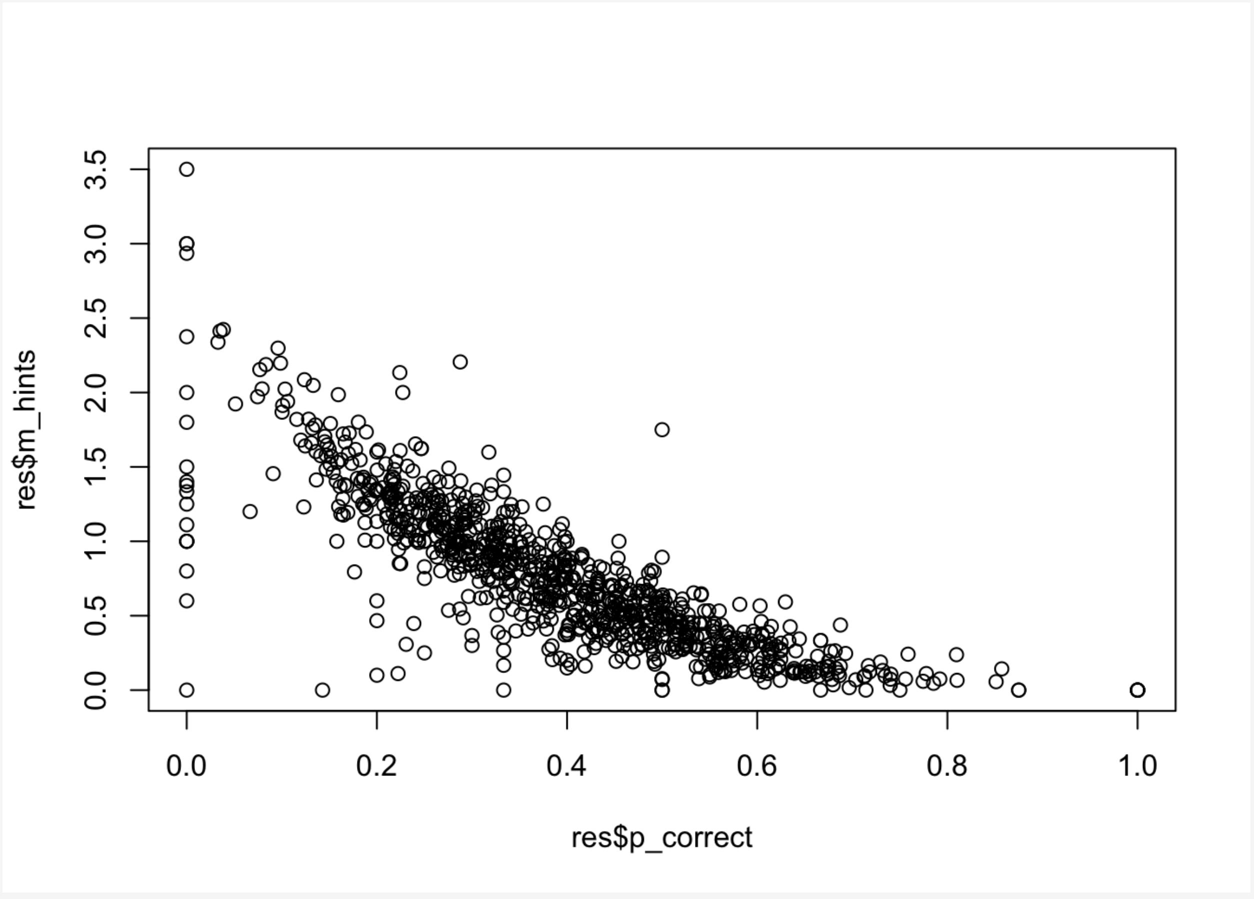
```
hist(res$p_correct)
```



```
#######################################
#######################################
```

Question 16: Do students who perform better ask for more hints? To answer, plot the student-level relationship (i.e. 1 value per student) between average correctness on first attempt (on the x-axis) and the average number of hints (on the y-axis) using a scatter plot. Tip: Use the `plot()` function.

```
#######################################
####### BEGIN INPUT: Question 16 ######
#######################################

res = asm %>%
  group_by(studentID) %>%
  summarise(
    p_correct = mean(correctonfirstattempt),
    m_hints = mean(hints)
  )

plot(res$p_correct, res$m_hints)
```



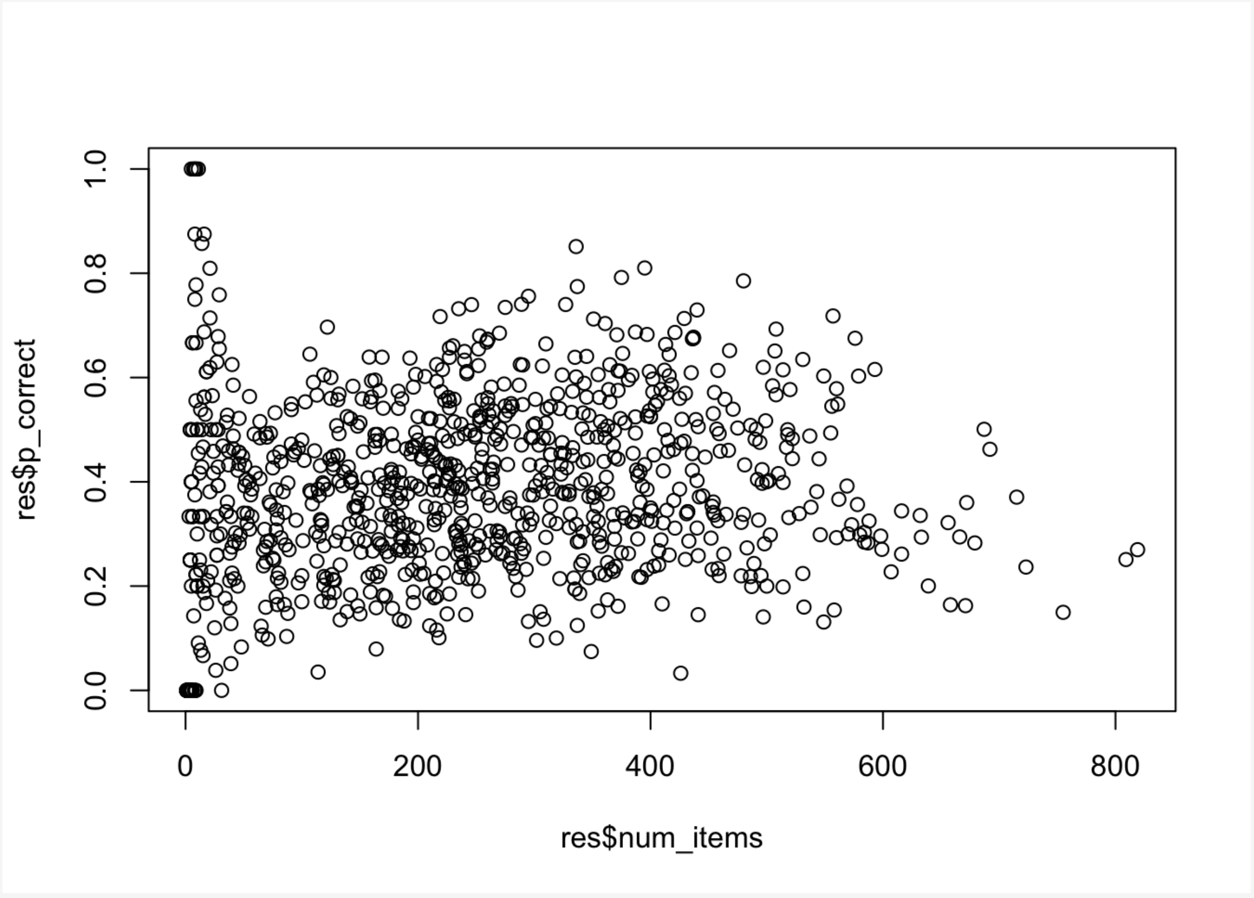
```
######################################
######################################
```

Question 17: Are students who answer more questions (i.e., get more practice) answering a larger proportion of questions correctly on the first attempt? To answer, report the correlation and make a scatterplot.

```
######################################
####### BEGIN INPUT: Question 17 ######
######################################

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

cor(res$num_items, res$p_correct)
```

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

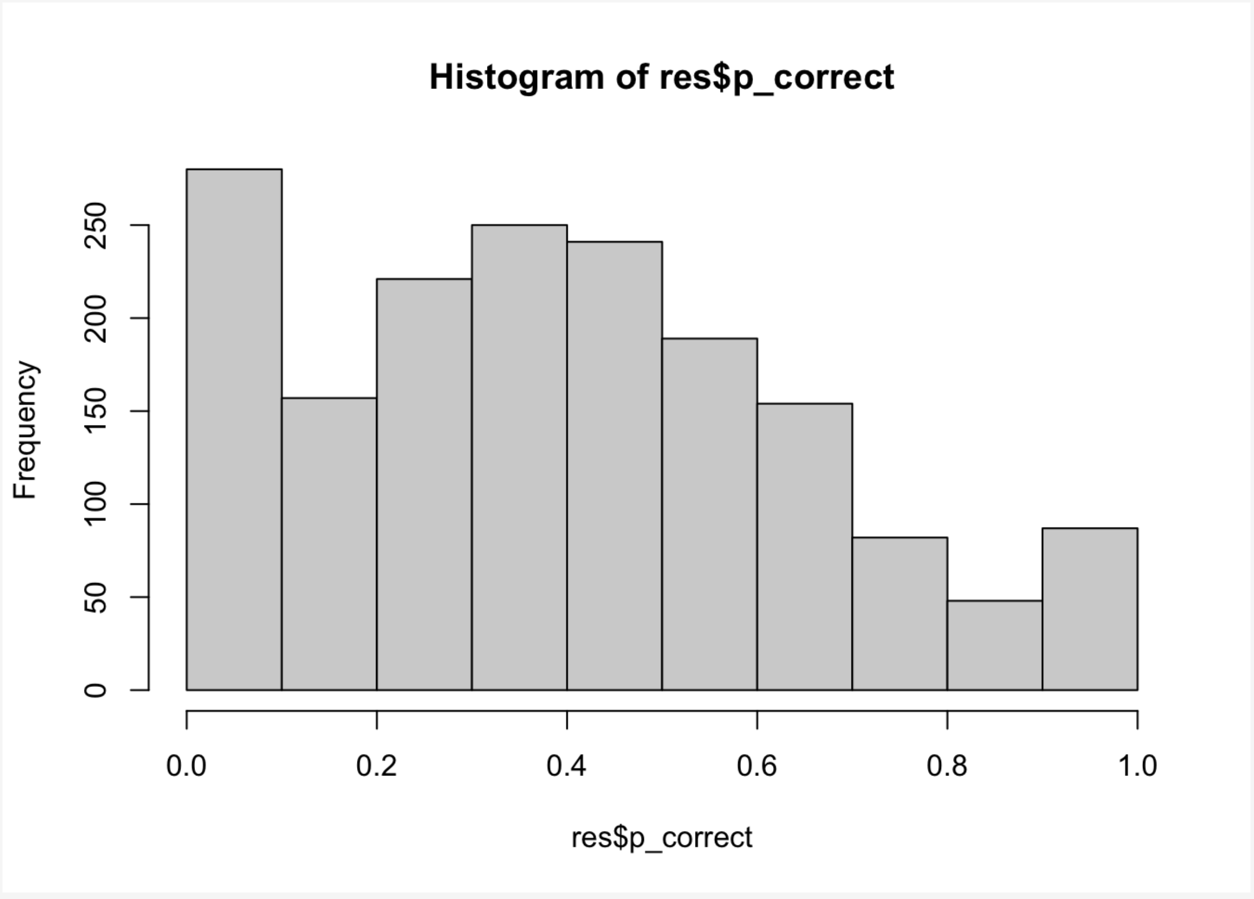
```
plot(res$num_items, res$p_correct)
```



```
#######################################
#######################################
```

Question 18:: How difficult are the questions? To answer, plot the question-level distribution (i.e. 1 row per question) of the proportion of students who get it right on the first attempt using a histogram. This quantity is also called `item difficulty’. Tip: Use the same general approach as for the student-level questions.

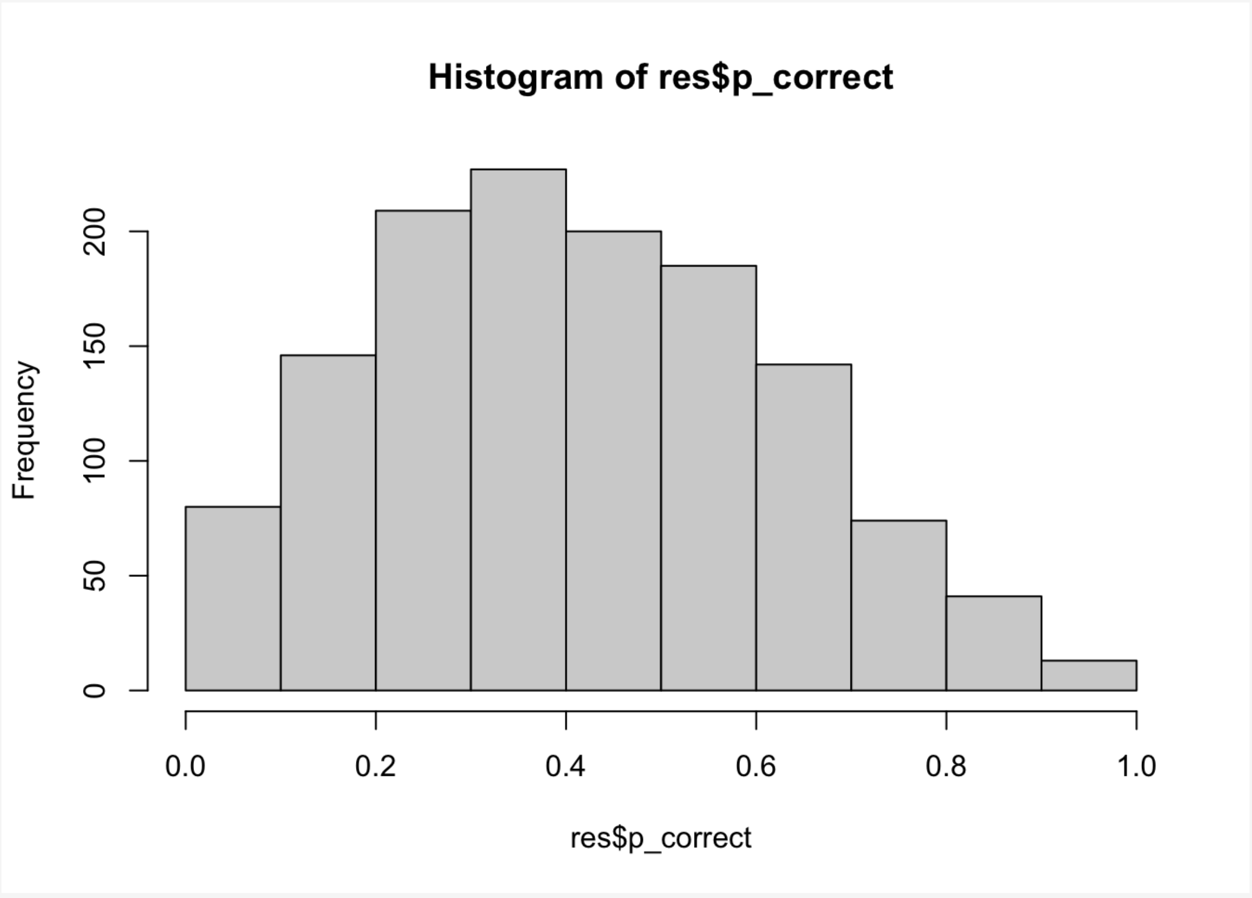
```
#######################################
####### BEGIN INPUT: Question 18 ######
#######################################

res = asm %>%
    group_by(itemid) %>%
    summarise(
        p_correct = mean(correctonfirstattempt)
    )

hist(res$p_correct)
```



```
########################################
########################################
```

Question 19: For the questions that were answered at least 10 times (i.e., where we have enough data to get a reliable estimate, here data from ten or more students), how difficult are the questions? Plot a histogram like before and note that you should see that the filtering reduces the spikes at 0 and 1). Tip: Use the `subset()` function in base R or `filter()` if using tidyverse.

```
########################################
####### BEGIN INPUT: Question 19 ######
########################################

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

hist(res$p_correct)
```



```
#######################################
#######################################
```

Self-reflection

Briefly summarize your experience on this homework. What was easy, what was hard, what did you learn?

- Insert your self-reflection here; it is used to improve the homework and course materials.

Estimate time spent

We want to give students an estimate of how much time this homework will take. Please indicate how many hours you spent to complete this homework here.

- I spent [insert your time] hours.

Generative AI usage

As stated in the course syllabus, using generative AI is allowed to help you as you complete this homework. We are interested in how it is being used and whether it is helpful for you.

- How much did you use generative AI (e.g., not at all, some, most, or all the questions) and which one did you use?
- If you used generative AI, how did you use it and was it helpful?

Submit Homework

This is the end of the homework. Please **Knit to Word**. The resulting file has to show both the R code and R output. Upload it on the EdX platform before the due date.