

Rote Memorization Techniques

W241 Summer 2020 (Daniel Hedblom Wed 6:30 pm) Final Project

Tina Huang, Gunnar Mein, Jeff Li

8/12/2020

Contents

Introduction	2
Background	2
Intuition	2
Hypothesis	2
Experiment design	3
Basic structure	3
Test subjects	3
Comparison of potential outcomes	3
Randomization process	4
Pre-treatment-phase content and test	4
Treatment-phase content and test	4
Power calculation	4
Analysis of results	5
Loading and preprocessing the data	5
EDA	5
Randomization checks	6
Effect calculation	9
Discussion of experiment	9
Suggestions for future experiments	11
Appendix	12
Code structure	12
Screen shots	13
Logs	17
Full analysis code	17

Introduction

Background

Rote memorization has a bad name in pedagogy, causing teachers to stay away from it and students to feel free to never practice it. However, in mathematics, physics, computer science, and other sciences, being able to recall certain snippets of knowledge quickly is imperative for both being able to perform calculations, for writing code in reasonable time, and to recognize patterns and form one's own "grammatical ability" to formulate approaches. It isn't really possible to imagine what a forest looks like, and how you would navigate it, until you can effortlessly recall what a tree is. Often, there are only a handful of vital concepts that a student needs to recall, but those they need to recall with absolutely no expenditure of time or doubt.

Teachers have, at times, built little structured exercises into their curricula to help students remember such snippets. One technique is for the student to read the fact/formula/code in question so that they have a memory of seeing the thing, writing it down in their own hand (or typing it, but I prefer hand-writing for this) so they have a memory of writing it, and lastly saying it out loud multiple times so they have memory of saying it.

This is all good and nice, but can we substantiate the validity of these techniques with an experiment? In other words, "is using multiple ways of remembering the same concept effective in aiding retention?"

Intuition

The justification for the memorization approach tested here comes mainly out of intuition. When we remember a line in a song, we don't remember just the words, we remember the sound of the vocalist singing them. When we remember a vacation at a beach, we remember the color of the sky and the sound of the waves and birds, not the abstract concepts "there were birds, the sky was blue". Instead, we reconstruct these abstract concepts from our memory of the experience of our senses.

When we need to memorize purely abstract concepts for their own sake, we can manufacture sense experiences by saying them out loud and writing them down (this might be less true for writing on a computer compared to paper and pencil, but stick with us here). It is our hope that these sense memories will be easier to recall, and aid the student in producing the right abstract facts to go along with them.

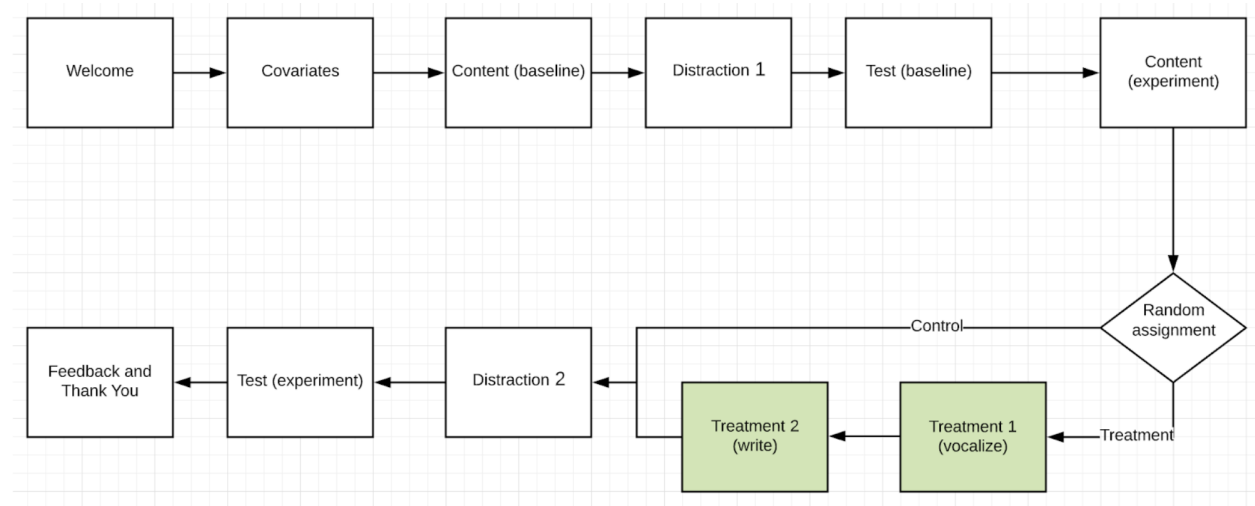
Hypothesis

Our specific hypothesis is, then, that students memorizing content by vocalizing them and writing them down should see a significantly higher score when evaluated over multiple such exercises, compared to students who only read the content quietly.

Experiment design

Basic structure

There were 2 portions of our test. The first portion was a “baseline” portion, where study participants would be presented 2 passages. The second portion would consist of the actual experiment, where our study participants would then be randomized into “treatment” or “control” groups.



Subjects were randomly assigned 2 topics in both Part 1 and Part 2 from a pool of 8 questions ranging from topics from “Units for the Calorie Content of Food” to the “History of the Submarine”. After reading each passage, they were given a video distraction and then given 4 multiple choice questions on what they had read. Screen shots are provided in the appendix with sample passages and questions that people were presented with

Test subjects

Since one of our team members is a high school teacher in Washington state, we decided to solicit the students of this school to participate in this experiment. Due to a need for additional study participants, all of the team members also solicited people on social media, as well as family and friends in order to participate in this experiment, rounding out our “considered” subjects. “Assigned” were subjects who at least made it as far as entering and submitting covariants. Since there was major attrition beyond that point, the “actually evaluated” subjects are only those who completed the whole test. We have detailed data about which subjects attrited at what point, and can compute an average treatment effect on the treated (ATET).

The actual age distribution of participant who completed the test is displayed in the results section. Middle school to high school age range accounted for around a third of the total participants. The rest were distributed between the 20 to 80 years old, and the older portion skews toward the female gender.

Comparison of potential outcomes

The format of our experiment was a multiple choice test. Study participants would read a passage for a predetermined amount of time (generally around 2 minutes per passage, depending on length), and would subsequently be presented with a set of multiple choice questions. Possible scores on each test ranges from 0 (fully incorrect) to 8 (fully correct). What we are interested in is the difference in score between the pre-test (before treatment) and the post-test (after treatment). For the control group, the expected difference between tests is 0 in an ideal world. In the real world, subjects might get tired between tests and score lower on the second part, or also might get warmed up to the process of memorization and testing and score higher. However, with proper randomization, the expected value of score difference between treatment and control should be 0 if the treatment has no effect.

Randomization process

Participants were assigned to “control” and “treatment” group randomly. The software did this right after covariates were gathered. In anticipation of collusion between students, we decided to cluster by time slot, in 5-minute increments. In hindsight, this was not necessary given our randomized draw from a question bank which made spillover effects unlikely, but it also did not hurt our process or errors too much. Students by-and-large took the tests in different time slots.

Pre-treatment-phase content and test

The first part of the experiment is the same for control and treatment - two content items are displayed along with questions we intend to ask, then a distraction takes the subject’s mind off the content for about a minute, and then we test the subject on the two topics with 4 questions per topic, with 5 possible choices for each question. This gives us 8 “correct”/“incorrect” scores, which we record as a “baseline” score.

Treatment-phase content and test

The treatment of this experiment is in Part 2. The participants in the treatment group will read 2 passages with 4 questions for each passage that they will answer later on. At the top of the screen, there is instruction to read the answers to the questions shown out loud 5 times. This allows the participants to have repeated memory of speaking the concept. On the screen after, the participants will be shown the same passages and questions. At the top of this screen, there is instruction for participants to read the passage again and type out brief answers to the questions. This allows the participants to have written memory of the concept. Showing the passages and instructing the participants to read the passages on 2 different screens allows them to have repeated memory of seeing the concept. The treatment as a whole allows the participants to have repeated memory of seeing and speaking the concept and written memory of the concept and therefore tests our hypothesis. After these 2 screens, the participants will watch a 1 minute distraction video, then go to a screen where they will select correct answers for the questions in multiple choice format. We once again record their 8 correct/incorrect answers, this time marked as “experiment” scores.

Power calculation

Prior to conducting our experiment, we used a ballpark estimate in order to estimate how many observations we would need. We used an online power calculator (<https://www.stat.ubc.ca/~rollin/stats/ssize/n2.html>) in order to estimate our necessary effect size. Given that each study participant would answer 8 questions (with 5 multiple choice options), we used a binomial distribution to estimate our value for σ (arriving at an estimated σ of 1.131).

We estimated an effect size of getting on average one additional question right between treatment and control. With the experiment parameters of 0.05 and power calculation of .8, we estimated that we would need roughly 21 samples per group.

In retrospect, this back of the envelope calculation was not extremely helpful. We would have benefited greatly from a pretest, however, due to a software bug, we were unable to properly assess the results of our pretest. In retrospect, our estimated effect size was far too high. People generally were able to score quite well on the exam regardless of if they were in treatment or control.

todo: how much power did we actually have?

Analysis of results

We ended the experiment at a time we set several days in advance, after discussing that new participation had ebbed off and that without further marketing, we could not expect more, and none of us had any grand ideas for producing more subjects. The day we picked lined up with our initial time line.

Loading and preprocessing the data

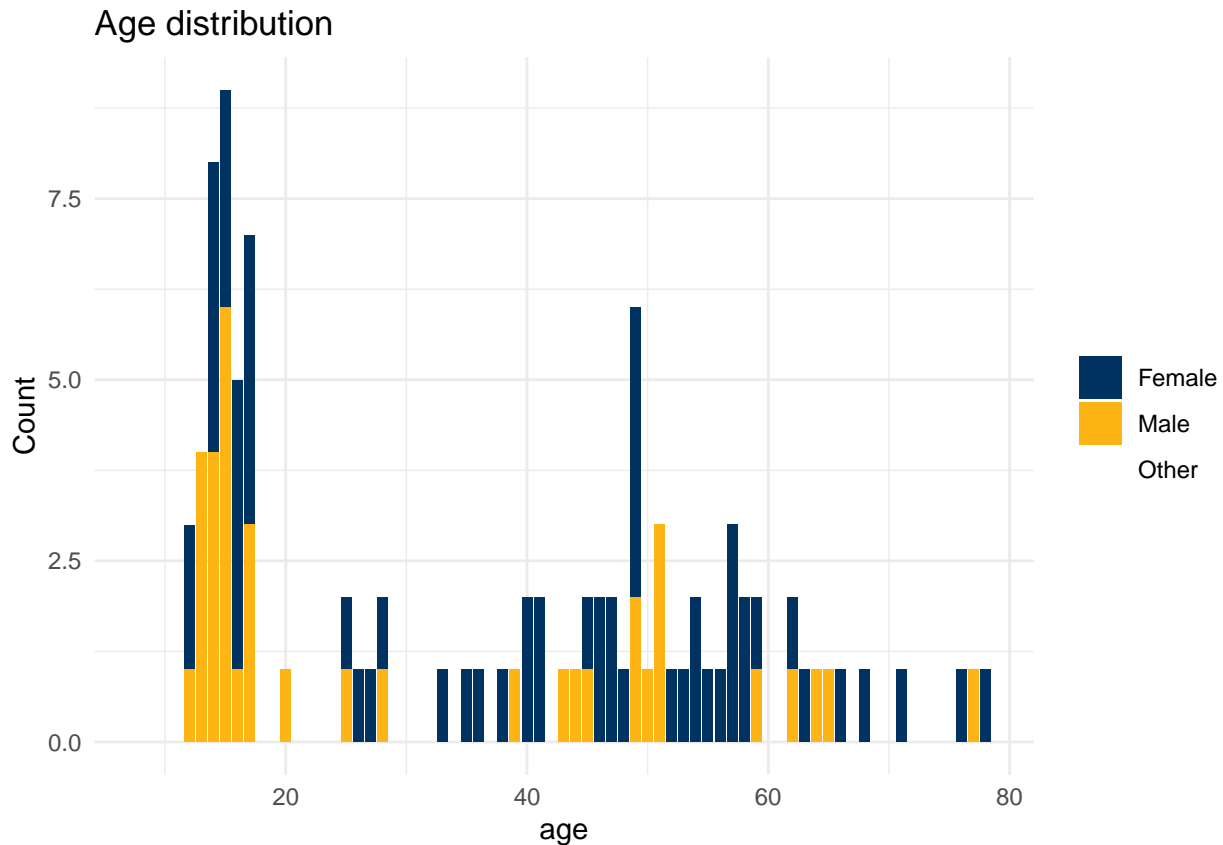
```
dataset = load_data()
dataset_control <- dataset %>% filter(treat == 0)
dataset_treat <- dataset %>% filter(treat == 1)
```

EDA

Table 1:

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
score	97	0.979	1.671	-2	0	2	5
score_pre	97	5.773	1.693	1	5	7	8
score_post	97	6.753	1.173	1	6	7	8
cluster	97	58.763	39.901	1	22	88	130
age	97	36.309	19.968	12	15	52	78
prior_knowledge	96	6.948	2.373	4.000	5.000	8.000	13.000
treat	97	0.485	0.502	0	0	1	1
reading	97	3.423	0.934	1	3	4	5
practice	97	2.278	0.933	1	1	3	5

At the end of the experiment, we had 50 subjects in control and 47 in treatment.



In the age distribution of only people who completed the whole experiment, we can see a group of Eastside Preparatory School students under the age of 20, a female-leaning cluster of certain Facebook friends over the age of 50, and a mixed group of other social media recruits between those groups. Gender balance does not look bad.

Randomization checks

Check by regression

We used 2 methods to validate the randomization process worked. The first method is to create a regression model using covariate information collected to predict whether the participant was assigned to control or treatment group. We created this regression model both for all participants who got assigned and for only participants who completed the survey. We concluded that the treatment assignment was random as none of the covariates had a significant predicting power. Details of the regression models are attached below.

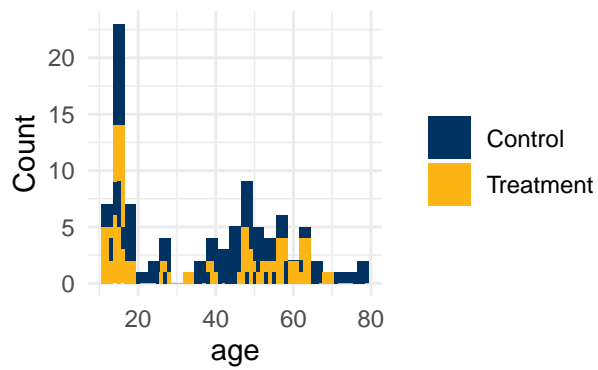
Visual check

The second method of validating randomization is to look at the distribution of control versus treatment assignment over participants' age, gender, self-reported memorization skill, and self-reported reading skill. As shown in the graphs below, the distribution of control versus treatment assignment are fairly even among all the participant groups. Therefore, we validated the randomization worked.

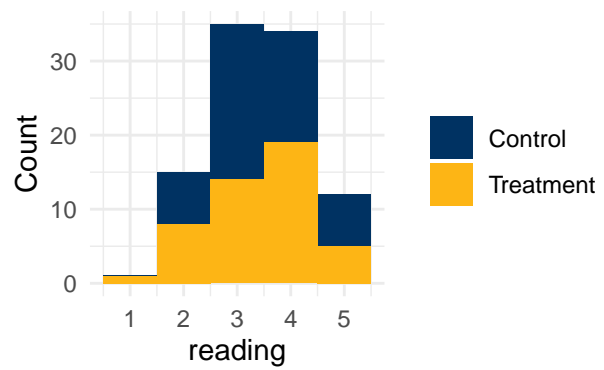
Table 2:

<i>Dependent variable:</i>	
	treat
age	−0.007 (0.011)
gender2	−0.318 (0.504)
practice	0.126 (0.247)
reading	−0.089 (0.242)
prior_knowledge	0.090 (0.104)
Constant	−0.332 (1.065)
Observations	94
Log Likelihood	−64.045
Akaike Inf. Crit.	140.090
<i>Note:</i>	*p<0.1; **p<0.05; ***p<0.01

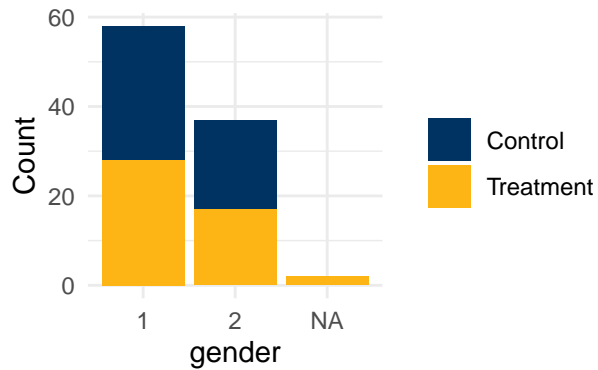
A Randomization – age



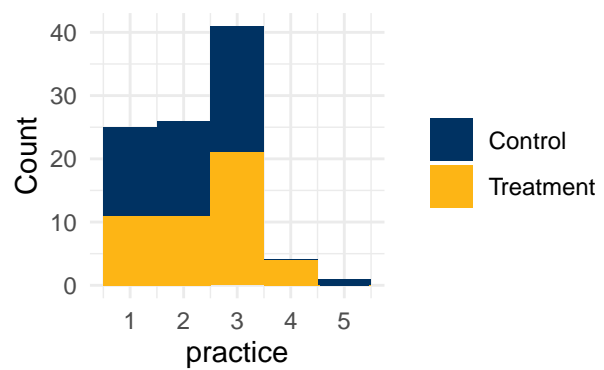
B Randomization – reading habits



C Randomization – gender

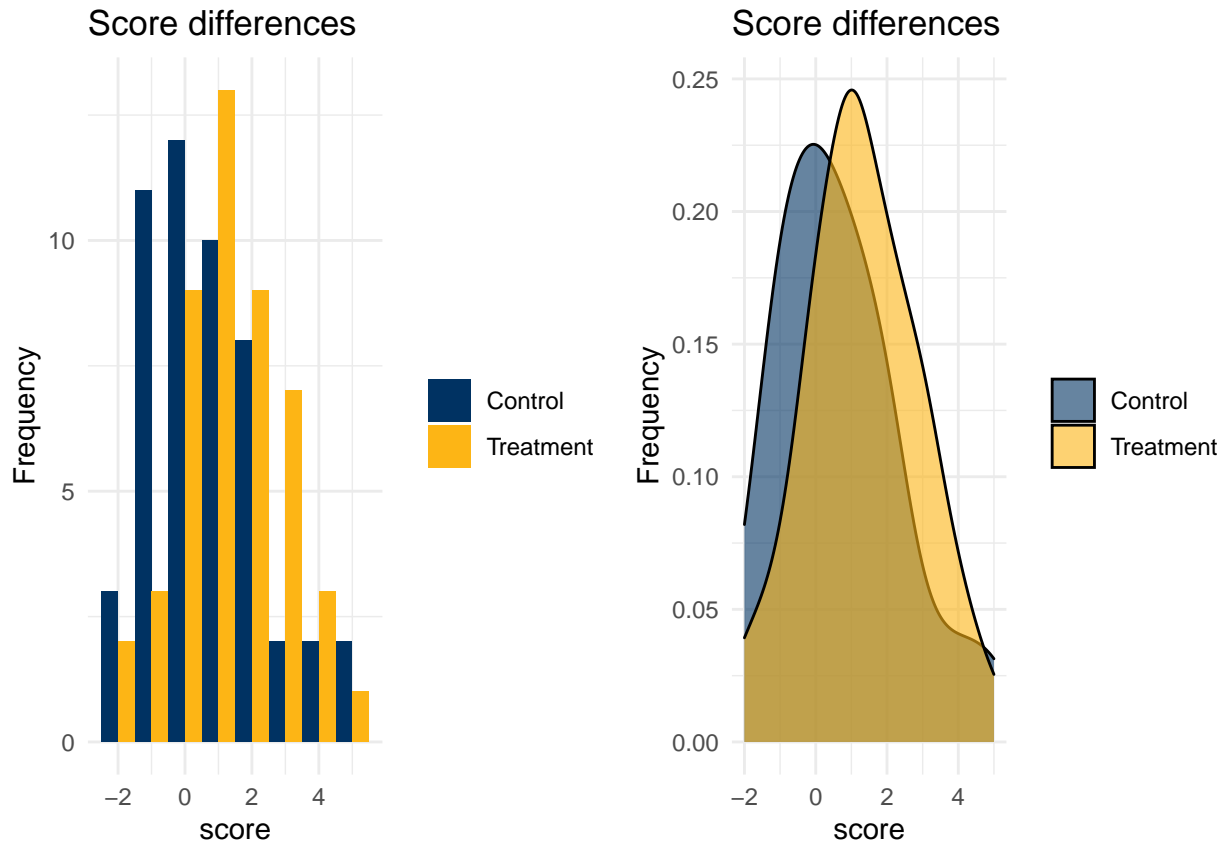


D Randomization – practice memoriz



Glance at score distribution

As part of our EDA, we take a look at the distribution of differential (post-pre) scores between treatment and control:



We can see the distribution shifted to the positive in the treatment group, by about a little under a full score point. Let's see whether it holds up statistically.

Effect calculation

As discussed above, one primary method of effect calculation for us was looking at the differences between pre- and post-treatment-phase scores. In this spirit, we now regress that difference on (1) only the treatment indicator `treat` and (2) on the treatment indicator along with a number of covariates to compute the ATET:

```
result_regressions()
```

Disappointingly, the inclusion of covariates did nothing to reduce our (cluster-robust) standard errors for the ATET.

todo: alternate calculation of treatment effect by panel?

Discussion of experiment

Attrition

todo

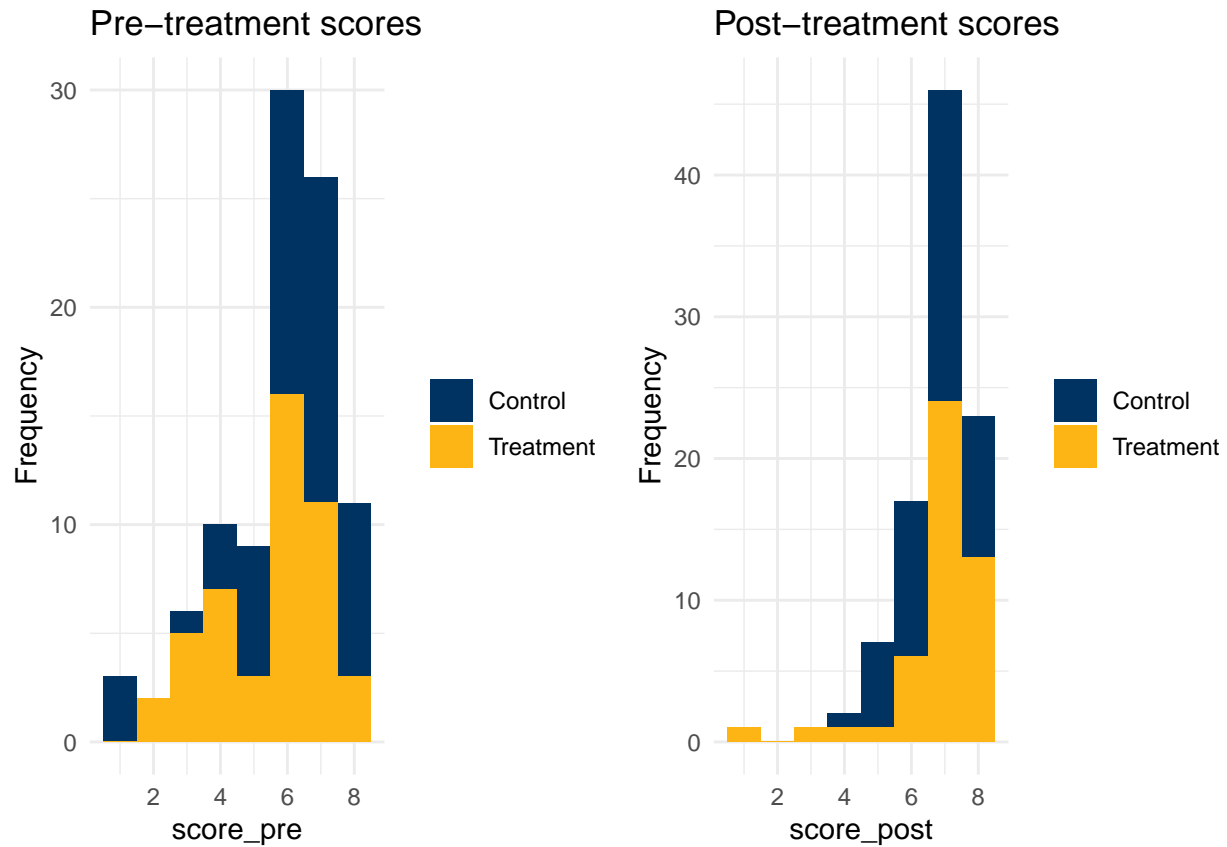
Calibration of questions / general scoring

A look at the pre- and post-treatment scores is instructive:

```
plot_scores()
```

Table 3: Regression Results with Clustered Standard Errors

	<i>Dependent variable:</i>	
	score	
	(1)	(2)
treat	0.659* (0.319)	0.663* (0.327)
age		0.006 (0.009)
prior_knowledge		−0.027 (0.077)
reading		−0.025 (0.184)
gender2		0.712 (0.399)
practice		0.153 (0.176)
Constant	0.660** (0.245)	0.126 (0.731)
Observations	97	94
Adjusted R ²	0.029	0.007
Residual Std. Error	1.646 (df = 95)	1.679 (df = 87)
F Statistic	3.884 (df = 1; 95)	1.112 (df = 6; 87)



Subjects scored well on our questions to begin with, there wasn't all that much room for improvement. This suggests that our content was too easy to memorize, or that our multiple-choice questions were too easy to answer.

Suggestions for future experiments

todo: expand

- Longer, bug-free pre-study
- Captive audience
- Harder questions
- Shorter content
- Hard-coded page flow

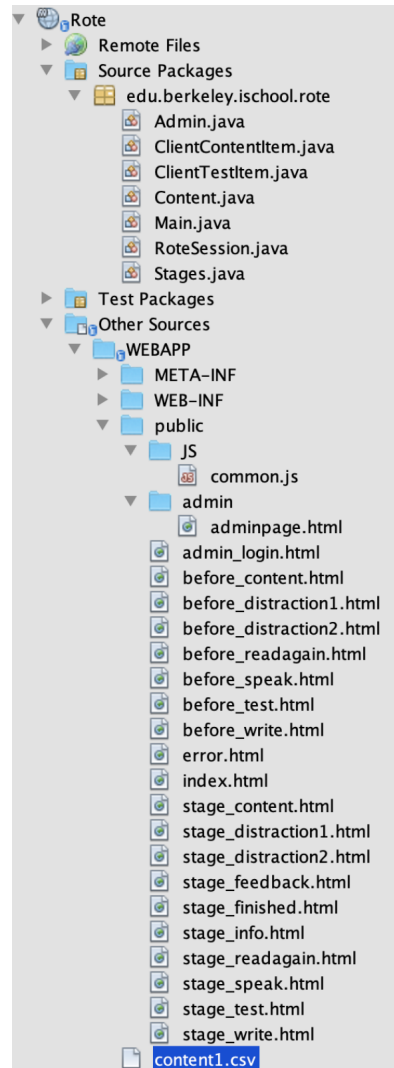
Appendix

Code structure

Rote is a Java project utilizing SparkJava as a bare-bones web-server framework. The client consists of plain HTML and JavaScript with a minimal dose of bootstrap CSS.

The project ran locally on a Tomcat server on MacOS for development and debugging, and on a similar Tomcat setup inside a Microsoft Azure Linux box for the real experiment. All code was developed in Apache NetBeans 11.3.

The repository is available at <https://github.com/gunnarmein-ucbischool/w241-rote>.



Screen shots

Welcome

Welcome to our survey!

Thank you for taking the time to help us today. This is an anonymous survey, we will never know who you are.

In this exercise, you will memorize answers to a total of 4 topics divided to part 1 and part 2. Please keep these in mind during the exercise:

1. This is not assessing your performance. Please complete this individually, we will share the combined results with you after August 12th.
2. There is a time limit, don't stress out over it, just do your best.
3. Please do not take notes unless instructed.
4. Take this survey only once. If a family member or friend wants to take the test on your computer, please open a new incognito browser window.

We hope you will enjoy memorizing some fun facts. Thank you for your contribution to our research :)

[Start survey](#) [Admin login](#)

Admin UI

[Download logs](#)

[Force control assignment](#) [Force treatment assignment](#) [Reset to random cluster assignment](#)

[Back to start page](#)

Live log:

```
Log: Content added: id: 2, title: The OSI Model
Log: Content added: id: 3, title: Famous potions from Harry Potter
Log: Content added: id: 4, title: Creation of Central Park
Log: Content added: id: 5, title: How money is made
Log: Content added: id: 6, title: History of the Submarine
Log: Content added: id: 7, title: Animal Migration
Log: Content added: id: 8, title: Units for the Calorie Content of Food
Log: Successfully read content csv /Users/gunnar/tomcat/webapps/Rote-1.0/content1.csv

----- start of log for new instance
Rote: Initializing
user.dir: /Users/gunnar/tomcat/bin
user.home: /Users/gunnar
Static files: /Users/gunnar/tomcat/webapps/Rote-1.0
System start time: 1596679080871
Rote: Fully initialized
-----
F2698E2CD140D490EE0BCCBBAD3A4B2C: Rote: New session, time: 1596907377018
```

Gathering covariates

DON'T GO BACK!!! The test will not work and your results will be lost

About You

Please tell us a tiny bit about yourself. This is all the information we will store, nothing can be associated with you. The more you can tell us, the more meaning our research will have.

Age:

Gender:

- ☐ Male
- ☐ Female
- ☐ Other
- ☐ Prefer not to say

How much do you read?

- ☐ Not at all
- ☐ Little
- ☐ Some
- ☐ A substantial amount
- ☐ All the time

How much practice do you have with memorization?

- ☐ None
- ☐ Little
- ☐ Some
- ☐ A substantial amount
- ☐ Expert

How much knowledge do you have about the topic "The OSI Model"?

- ☐ None

Intermission (one example only)

DON'T GO BACK!!! The test will not work and your results will be lost

Coming up: Memorization

Thank you!

You will see 2 passages of text, with 5 questions each. The objective is to be able to answer the questions.
Glance at the questions, read the passages, read the questions again.
Memorize the answers.

Press "Continue" when ready ...

Continue

Memorization

Memorization

DON'T GO BACK!!! The test will not work and your results will be lost!

Memorization

6:20

Please memorize the answer to the questions for each of the following content passages:

The OSI Model

The Open Systems Interconnection (OSI) Model breaks down network communication into seven layers. It's based on the concept of splitting up a communication system into seven abstract layers, each one stacked upon the last. In plain English, the OSI provides a standard for different computer systems to be able to communicate with each other.

What are the seven layers of the OSI model? The seven abstraction layers of the OSI model can be defined as follows, from top to bottom:

7. The Application Layer

This is the only layer that directly interacts with data from the user. Software applications like web browsers and email clients rely on the application layer to initiate communications. Application layer protocols include HTTP as well as SMTP (Simple Mail Transfer Protocol is one of the protocols that enables email communications).

- Question: Which 2 layers sit between the Application Layer and the Transport Layer?
- Question: Based on the article, what layer is responsible for flow control and error control?
- Question: How many layers comprise the OSI Model?
- Question: What does OSI stand for?

Creation of Central Park

Before Central Park was created, the landscape along what is now the Park's perimeter from West 82nd to West 89th Street was the site of Seneca Village, a community of predominantly African-Americans, many of whom owned property. Seneca Village began in 1825, by the mid 1850s, the village consisted of approximately 225 residents, made up of roughly two-thirds African-Americans, one-third Irish immigrants, and a small number of individuals of German descent. Compared to other African-Americans living in New York, residents of Seneca Village seem to have been more stable and prosperous. By 1855, approximately half of them owned their own homes. With property ownership came other rights not commonly held by African-Americans in the City, namely, the right to vote. In 1821, New York State required African-American men to own at least \$250 in property and hold residency for at least three years to be able to vote. Of the 100 black New Yorkers eligible to vote in 1845, 10 lived in Seneca Village.

During the early 1850s, the City began planning for a large municipal park to counter unhealthy urban conditions and provide space for recreation. In 1853, the New York State Legislature enacted a law that set aside 775 acres of land in

- Question: By 1855, what was the Seneca village made up of roughly?
- Question: In 1821, New York State required African-American men to meet which requirement(s) to vote?
- Question: What is eminent domain law?
- Question: Of the 100 black New Yorkers eligible to vote in 1845, how many lived in Seneca Village?

Treatment (part 1)

DON'T GO BACK!!! The test will not work and your results will be lost!

Technique: Speaking out loud!

3:16

Please speak the answer to each of the questions out loud 5 times before you continue.

Famous potions from Harry Potter

Below are some famous potions from the series Harry Potter:

Felix Felicis/Liquid Luck

Felix Felicis, more commonly known as Liquid Luck, grants whoever drinks it unusually good luck. The time span of this luck depends on the amount imbibed.

Pepperup Potion

A Pepperup Potion is designed to improve health, relieve coughs and colds, though it does have one major side effect: it causes steam to dribble from the patient's ears for several hours afterward. It is also used to quickly elevate body temperature, as shown after the second task in Harry Potter and the Goblet of Fire and mentioned in Harry Potter and the Chamber of Secrets.

- Question: Which of the following is a side effect from the Pepperup Potion?
- Question: What is the other name for Felix Felicis?
- Question: Which of the following is not an ingredient for Polyjuice Potion?
- Question: What does the Polyjuice potion do?

How money is made

The ordinary paper that consumers use throughout their everyday life such as newspapers, books, cereal boxes, etc., is primarily made of wood pulp; however, United States currency paper is composed of 75 percent cotton and 25 percent linen. This is what gives the United States currency its distinct look, feel and durability. Bills over \$1 have red and blue strands woven in for security purposes. Dollar bills do not have this feature.

All physical monetary production is overseen by the Bureau of Engraving and Printing, which has printing facilities in Washington, D.C. and Fort Worth, Texas. The image of the dollar bill is engraved onto giant steel plates. These sheets of soft steel are then curved slightly and bolted onto a printing cylinder. Ink is added and then wiped off, leaving only ink in the engraved, recessed areas of the plate. The special sheets of paper, together in massive rolls, are then fed through the machines as they rotate, exerting 20 tons of pressure from the top and the bottom. These machines can print around 8,000 sheets per hour. Sheets of bills then pass through a state-of-the-art computer scanning process that checks for inconsistencies. When complete, bills are packaged and shipped all over the country.

- Question: What is the United States currency paper is composed of?
- Question: Bills over what dollar amount have red and blue strands woven in for security purposes?
- Question: Where are the U.S. paper currency printing facilities?
- Question: How many sheets can bill-printing machines print per hour?

Treatment (part 2)

DON'T GO BACK!!! The test will not work and your results will be lost!

Technique - Write it down!

3:17

Please look at the content again, and write down your brief answers in the boxes:

Famous potions from Harry Potter

Below are some famous potions from the series Harry Potter:

Felix Felicis/Liquid Luck

Felix Felicis, more commonly known as Liquid Luck, grants whoever drinks it unusually good luck. The time span of this luck depends on the amount imbibed.

Pepperup Potion

A Pepperup Potion is designed to improve health, relieve coughs and colds, though it does have one major side effect: it causes steam to dribble from the patient's ears for several hours afterward. It is also used to quickly elevate body temperature, as shown after the second task in Harry Potter and the Goblet of Fire and mentioned in Harry Potter and the Chamber of Secrets.

Question: Which of the following is a side effect from the Pepperup Potion?

Answer:

Question: What is the other name for Felix Felicis?

Answer:

Question: Which of the following is not an ingredient for Polyjuice Potion?

Answer:

Question: What does the Polyjuice potion do?

Answer:

How money is made

The ordinary paper that consumers use throughout their everyday life such as newspapers, books, cereal boxes, etc., is primarily made of wood pulp; however, United States currency paper is composed of 75 percent cotton and 25 percent linen. This is what gives the United States currency its distinct look, feel and durability. Bills over \$1 have red and blue strands woven in for security purposes. Dollar bills do not have this feature.

All physical monetary production is overseen by the Bureau of Engraving and Printing, which has printing facilities in Washington, D.C. and Fort Worth, Texas. The image of the dollar bill is engraved onto giant steel plates. These sheets of soft steel are then curved slightly and bolted onto a printing cylinder. Ink is added and then wiped off, leaving only ink in the engraved, recessed areas of the plate. The special sheets of paper, together in massive rolls, are then fed through the machines as they rotate, exerting 20 tons of pressure from the top and the bottom. These machines can print around 8,000 sheets per hour.

Sheets of bills then pass through a state-of-the-art computer scanning process that checks for inconsistencies. When complete, bills are packaged and shipped all over the country.

Question: What is the United States currency paper is composed of?

Answer:

Question: Bills over what dollar amount have red and blue strands woven in for security purposes?

Answer:

Question: Where are the U.S. paper currency printing facilities?

Answer:

Question: How many sheets can bill-printing machines print per hour?

Answer:

Test

DON'T GO BACK!!! The test will not work and your results will be lost!

Time to put your memory to the test!

6:24

The OSI Model:

Which 2 layers sit between the Application Layer and the Transport Layer?

- ☐ Presentation Layer, Session Layer
- ☐ Data Link Layer, Presentation Layer
- ☐ Transmission Layer, Network Layer
- ☐ Data Link Layer, Physical Layer
- ☐ None of the above
- ☐ I don't remember

Based on the article, what layer is responsible for flow control and error control?

- ☐ Presentation Layer
- ☐ Data Link Layer
- ☐ Transport Layer
- ☐ The Network Layer
- ☐ The Session Layer
- ☐ I don't remember

How many layers comprise the OSI Model?

- ☐ 5
- ☐ 6
- ☐ 7
- ☐ 8
- ☐ 9
- ☐ I don't remember

What does OSI stand for?

- ☐ Operating System Interface
- ☐ Open System Internet
- ☐ Open Systems Interconnection
- ☐ Operating Systems Interconnect
- ☐ None of the above
- ☐ I don't remember

Creation of Central Park:

By 1855, what was the Seneca village made up of roughly?

- ☐ two-thirds Irish, one-third African-Americans immigrants
- ☐ two-thirds African-Americans, one-third Irish immigrants
- ☐ two-thirds African-Americans, one-third German immigrants
- ☐ two-thirds German, one-third African-Americans immigrants
- ☐ None of the above
- ☐ I don't remember

In 1821, New York State required African-American men to meet which requirement(s) to vote?

- ☐ Own property and hold residency for at least five years
- ☐ Own at least \$250 in property and hold residency for at least three years
- ☐ Own at least \$250 in property and have no criminal record
- ☐ Have paid at least \$250 in property tax and hold residency for at least five years
- ☐ None of the above
- ☐ I don't remember

What is eminent domain law?

- ☐ The law allows the government to take private land for public use with compensation paid to the landowner
- ☐ The law allows the government to take private land for public use by providing alternative property to the landowner
- ☐ The law allows the government to take private land for public use by providing tax credit to the landowner
- ☐ The law allows the government to take private land for public use by sharing ownership of the new establishment with the landowner
- ☐ None of the above
- ☐ I don't remember

Of the 100 black New Yorkers eligible to vote in 1845, how many lived in Seneca Village?

- ☐ 25
- ☐ 15
- ☐ 10
- ☐ 45
- ☐ None of the above
- ☐ I don't remember

Feedback

DON'T GO BACK!!! The test will not work and your results will be lost!

Feedback

Do you have any feedback on your experience? Anything we should improve?

(500 characters or fewer, please)

Submit and finish

Logs

The general operations log, covariates log, written submissions log, and test results log can be viewed at:
https://drive.google.com/drive/folders/1t1wtW139ow35JUcuspJsWoYNjJzNNaK_?usp=sharing

Full analysis code

```
load_data = function(){
  # import files
  rote_cov_original <- read.csv(file="rote_cov.csv", sep=',') #covariates
  rote_test_original <- read.csv(file="rote_test.csv", sep=',') #test file

  # create copies
  rote_cov <- rote_cov_original
  rote_test <- rote_test_original

  # convert unix epoch time to datetime
  # cov
  rote_cov$session_start_time <- as.POSIXct(rote_cov$session_start_time/1000, origin="1970-01-01")
  rote_cov$cov_submit_time <- as.POSIXct(rote_cov$cov_submit_time/1000, origin="1970-01-01")

  # test
  rote_test$session_start_time <- as.POSIXct(rote_test$session_start_time/1000, origin="1970-01-01")
  rote_test$test_submit_time <- as.POSIXct(rote_test$test_submit_time/1000, origin="1970-01-01")
  rote_test$test_time <- rote_test$test_submit_time - rote_test$session_start_time

  # remove rows with any NAs in the answers(indicates they did not complete the test)
  rote_test <- rote_test %>% drop_na("a11", "a12", "a13", "a14", "a21", "a22", "a23","a24")

  rote_test_baseline <- rote_test %>% filter(test == 'baseline')
  rote_test_experiment <- rote_test %>% filter(test == 'experiment') %>%
    select('session_id','item_id1','a11','a12', 'a13', 'a14', 'a21', 'a22', 'a23','a24',
           'item_id2', 'c11','c12', 'c13', 'c14', 'c21', 'c22', 'c23','c24')

  rote_test <- inner_join(rote_test_baseline, rote_test_experiment, by="session_id" )

  # remove rows with missing covariates
  # convert age, remove outliers for age
  suppressWarnings(
    rote_cov$age <- ifelse(as.numeric(as.character(rote_cov$age)) > 100,
                          NA, as.numeric(as.character(rote_cov$age)))
  )
  rote_cov <- rote_cov %>% drop_na("age")

  #fix gender
  rote_cov$gender <- ifelse(rote_cov$gender == 'other', NA, rote_cov$gender)

  # on test file, establish if testee answer matches actual answer.
  # if it matches, set var to 1, else 0.
  # notation will be o11, o12, etc (o is for outcome)
  # x = baseline
  # y = treatment
```

```

rote_test$o11x <- ifelse(rote_test$a11.x == rote_test$c11.x, 1, 0)
rote_test$o12x <- ifelse(rote_test$a12.x == rote_test$c12.x, 1, 0)
rote_test$o13x <- ifelse(rote_test$a13.x == rote_test$c13.x, 1, 0)
rote_test$o14x <- ifelse(rote_test$a14.x == rote_test$c14.x, 1, 0)
rote_test$o21x <- ifelse(rote_test$a21.x == rote_test$c21.x, 1, 0)
rote_test$o22x <- ifelse(rote_test$a22.x == rote_test$c22.x, 1, 0)
rote_test$o23x <- ifelse(rote_test$a23.x == rote_test$c23.x, 1, 0)
rote_test$o24x <- ifelse(rote_test$a24.x == rote_test$c24.x, 1, 0)

rote_test$o11y <- ifelse(rote_test$a11.y == rote_test$c11.y, 1, 0)
rote_test$o12y <- ifelse(rote_test$a12.y == rote_test$c12.y, 1, 0)
rote_test$o13y <- ifelse(rote_test$a13.y == rote_test$c13.y, 1, 0)
rote_test$o14y <- ifelse(rote_test$a14.y == rote_test$c14.y, 1, 0)
rote_test$o21y <- ifelse(rote_test$a21.y == rote_test$c21.y, 1, 0)
rote_test$o22y <- ifelse(rote_test$a22.y == rote_test$c22.y, 1, 0)
rote_test$o23y <- ifelse(rote_test$a23.y == rote_test$c23.y, 1, 0)
rote_test$o24y <- ifelse(rote_test$a24.y == rote_test$c24.y, 1, 0)

rote_test$score_pre <- rote_test$o11x + rote_test$o12x + rote_test$o13x +
  rote_test$o14x + rote_test$o21x + rote_test$o22x + rote_test$o23x + rote_test$o24x
rote_test$score_post <- rote_test$o11y + rote_test$o12y + rote_test$o13y +
  rote_test$o14y + rote_test$o21y + rote_test$o22y + rote_test$o23y + rote_test$o24y

rote_test$score = rote_test$score_post - rote_test$score_pre

#on covariates file, sum knowledge scores together
rote_cov$knowledge_cov_pre <- rote_cov$knowledge1 + rote_cov$knowledge2
rote_cov$knowledge_cov_post <- rote_cov$knowledge3 + rote_cov$knowledge4
rote_cov$prior_knowledge <- rote_cov$knowledge_cov_post + rote_cov$knowledge_cov_pre

#convert necessary columns to boolean
rote_test$treat <- ifelse(rote_test$treat == "false", 0, 1)
rote_cov$treat <- ifelse(rote_cov$treat == "false", 0, 1)

#get distinct ids
rote_test <- rote_test[!duplicated(rote_test$session_id),]
rote_cov <- rote_cov[!duplicated(rote_cov$session_id),]

#inner join 2 datasets
dataset <- inner_join(rote_test, rote_cov, by="session_id" )

dataset$treat <- dataset$treat.x
dataset$cluster <- dataset$cluster.x
dataset$gender <- as.factor(dataset$gender)

dataset <- dataset %>%
  filter %>%
  select("session_id", "score", "score_pre", "score_post", "cluster", "gender", "age",
    "prior_knowledge", "treat", "reading", "practice")
}

attrition = function() {
  # todo: df3 is the dataset before covariates are joined, need to see how to reconstruct that

```

```

df2 = left_join(df_b, df_e, by='session_id')
nrow(df2)
df3 = left_join(df_cov, df2, by='session_id')
nrow(df3)
df3a= inner_join(df_b, df_e, by="session_id")
df_completed = inner_join(df_cov, df3a, by="session_id")
nrow(df_completed)

df3$completed = as.factor(ifelse(!is.na(df3$a11.y), "completed",
                                ifelse(!is.na(df3$a11.x), "baseline",
                                ifelse(!is.na(df3$treat),
                                "covariates", "started"))))

summary(df3$completed)
df3$treat = as.numeric(df3$treat)
str(df3)
df3$completed = factor(df3$completed,
                      levels=c("started", "covariates", "baseline", "completed"))

str(df3)
n=218-171
df_started = data.frame(session_id = as.factor(1:n), treat=3, completed=factor("started"))
str(df_started)
df4 = full_join(df3, df_started, by="session_id")
str(df4)
df4$treat = ifelse(is.na(df4$treat.x), df4$treat.y.y, df4$treat.x)
df4$completed = as.factor(ifelse(is.na(df4$completed.x), df4$completed.y, df4$completed.x))
}

print_stats = function() {
  print(paste("After cleaning, the number of rows in our dataset is:",
              toString(nrow(dataset))))
  print(paste("After cleaning, the number of observations in treatment is:",
              toString(sum(dataset$treat))))
  print(paste("After cleaning, the number of observations in control is:",
              toString(nrow(dataset) - sum(dataset$treat))))
}

print_summary = function() {
  stargazer(dataset,
            header= F,
            title = "Summary Table of Data",
            type="latex") #flip type between text and latex
}

result_regressions = function() {
  regression1 <- lm(score ~ treat ,data=dataset)
  regression2 <- lm(score ~ treat + age + prior_knowledge + reading + gender + practice,data=dataset)
  clustered_errors1 <- vcovCL_1c <- vcovCL(regression1, cluster = dataset[, 'cluster'])
  clustered_errors2 <- vcovCL_2c <- vcovCL(regression2, cluster = dataset[, 'cluster'])
  stargazer(regression1, regression2,
            header = F,
            type = "latex",

```

```

    omit.table.layout= "n",
    keep.stat = c("adj.rsq", "n", "f", "ser", "aic", "wald"),
    se = list(sqrt(diag(clustered_errors1)),sqrt(diag(clustered_errors2))),
    star.cutoffs = c(0.05, 0.01, 0.001),
    title="Regression Results with Clustered Standard Errors")
}

cov_check_regression = function() {
  glm(treat~age+gender+practice+reading+prior_knowledge,
      data=dataset, family=binomial(link="logit"))
}

plot_rand_gender = function (){
  dataset %>% ggplot()+
    geom_bar(aes(x=gender, fill=as.factor(treat)),
             position="stack") +
    theme_minimal() +
    #scale_fill_brewer(palette="Dark2") +
    scale_fill_manual(values=c("#003262", "#FDB515"),
                      labels=c("Control", "Treatment"))+
    ylab("Count") +
    ggtitle("Randomization - gender") +

    theme(legend.title=element_blank())
}

plot_rand_age = function (){
  dataset %>%
    ggplot(aes(x=age, fill=as.factor(treat)))+
    geom_histogram(position="stack", binwidth=3) +
    stat_count()+
    theme_minimal() +
    #scale_fill_brewer(palette="Dark2") +
    scale_fill_manual(values=c("#003262", "#FDB515"),
                      labels=c("Control", "Treatment"))+
    ylab("Count") +
    ggtitle("Randomization - age") +

    theme(legend.title=element_blank())
}

plot_rand_reading = function (){
  dataset %>% ggplot(aes(x=reading, fill=as.factor(treat)))+
    geom_histogram(position="stack", binwidth=1) +
    stat_count()+
    theme_minimal() +
    #scale_fill_brewer(palette="Dark2") +
    scale_fill_manual(values=c("#003262", "#FDB515"),
                      labels=c("Control", "Treatment"))+
    ylab("Count") +
    ggtitle("Randomization - reading habits") +

    theme(legend.title=element_blank())
}

```

```

plot_rand_practice = function () {
  dataset %>% ggplot(aes(x=practice, fill=as.factor(treat)))+
    geom_histogram(position="stack", binwidth=1) +
    stat_count()+
    theme_minimal() +
    #scale_fill_brewer(palette="Dark2") +
    scale_fill_manual(values=c("#003262", "#FDB515"),
                      labels=c("Control", "Treatment"))+
    ylab("Count") +
    ggtitle("Randomization - practice memorizing") +

    theme(legend.title=element_blank())
}

plot_scores = function() {
  dataset %>% ggplot(aes(x=score, fill=as.factor(treat)))+
    geom_histogram(position="dodge") +
    stat_count()+
    theme_minimal() +
    #scale_fill_brewer(palette="Dark2") +
    scale_fill_manual(values=c("#003262", "#FDB515"),
                      labels=c("Control", "Treatment"))+
    ylab("Frequency") +
    ggtitle("Score difference between baseline and experiment") +
    theme(legend.title=element_blank())
}

plot_gender = function() {
  ggplot(dataset, aes(x=as.factor(gender))) +
  geom_bar() +
  ggtitle("Distribution of Gender")
}

plot_age = function() {
  dataset %>% ggplot(aes(x=age, fill=gender))+
    geom_bar(position="stack") +
    stat_count()+
    theme_minimal() +
    #scale_fill_brewer(palette="Dark2") +
    scale_fill_manual(values=c("#003262", "#FDB515", "#C4820E"),
                      labels=c("Female", "Male", "Other"))+
    ylab("Count") +
    ggtitle("Age distribution") +
    theme(legend.title=element_blank())
}

plot_score_diffs = function () {
  dataset %>% ggplot(aes(x=score, fill=as.factor(treat)))+
    geom_histogram(position="dodge", binwidth=1) +
    theme_minimal() +
    scale_fill_manual(values=c("#003262", "#FDB515"),
                      labels=c("Control", "Treatment"))+
    ylab("Frequency") +

```

```

    ggtitle("Score differences") +
    theme(legend.title = element_blank())
}

plot_score_diffs_smooth = function () {
  dataset %>% ggplot(aes(x=score, fill=as.factor(treat)))+
    geom_density(alpha=0.6)+
    theme_minimal() +
    scale_fill_manual(values=c("#003262", "#FDB515"),
                      labels=c("Control", "Treatment"))+
    ylab("Frequency") +
    ggtitle("Score differences") +
    theme(legend.title = element_blank())
}

plot_score_pre = function () {
  dataset %>% ggplot(aes(x=score_pre, fill=as.factor(treat)))+
    geom_histogram(position="stack", binwidth=1) +
    theme_minimal() +
    scale_fill_manual(values=c("#003262", "#FDB515"),
                      labels=c("Control", "Treatment"))+
    ylab("Frequency") +
    ggtitle("Pre-treatment scores") +
    theme(legend.title = element_blank())
}

plot_score_post = function () {
  dataset %>% ggplot(aes(x=score_post, fill=as.factor(treat)))+
    geom_histogram(position="stack", binwidth=1) +
    theme_minimal() +
    scale_fill_manual(values=c("#003262", "#FDB515"),
                      labels=c("Control", "Treatment"))+
    ylab("Frequency") +
    ggtitle("Post-treatment scores") +
    theme(legend.title = element_blank())
}

plot_scores = function() {
  p1 = plot_score_pre()
  p2 = plot_score_post()
  ggarrange(p1, p2, nrow=1, ncol=2)
}

plot_diffs_all = function() {
  p1 = plot_score_diffs()
  p2 = plot_score_diffs_smooth()
  ggarrange(p1, p2, nrow=1, ncol=2)
}

plot_participation = function() {
  df_a = data.frame(id=1:4, stage=c("started", "covariates", "baseline", "complete"),
                    n=c(218, 171, 123, 108))
  df_a %>% ggplot(aes(x=stage, y=n, fill=stage))+

```

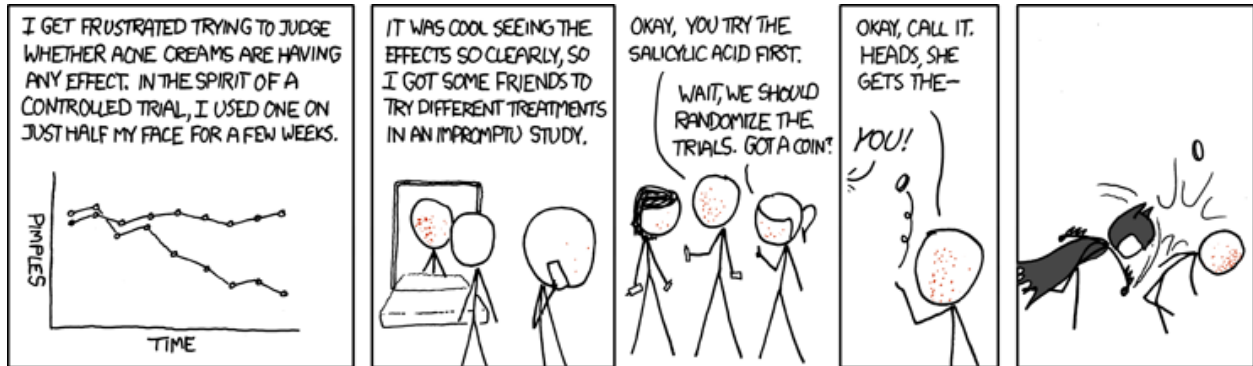
```

    geom_bar(stat="identity") +
    theme_minimal() +
    scale_x_discrete(limits=c("started", "covariates", "baseline", "complete")) +
    scale_fill_manual(values=c("#003262", "#3B7EA1", "#FDB515", "#C4820E")) +
    ylab("Participants") +
    ggtitle("Participation by stage")
}

plot_attrition = function () {
  df4 %>%
    filter(completed != 4) %>%
    ggplot(aes(x=completed, fill=as.factor(treat)))+
    geom_histogram(stat="count") +
    theme_minimal() +
    scale_x_discrete(labels=c("started", "covariates", "baseline", "complete")) +
    scale_fill_brewer(palette="Dark2") +
    scale_fill_manual(values=c("#003262", "#3B7EA1", "#FDB515", "#C4820E"),
                      labels=c("Control", "Treatment", "Before assignment")) +
    ylab("Participants dropping out") +
    ggtitle("Attrition by stage") +
    theme(legend.title=element_blank())
}

plot_rand = function() {
  p1 = plot_rand_age()
  p2 = plot_rand_reading()
  p3 = plot_rand_gender()
  p4 = plot_rand_practice()
  ggarrange(p1,p2,p3,p4,
            labels = c("A","B","C","D"),
            ncol = 2, nrow = 2)
}

```



<https://xkcd.com/700/>