Rote Memorization Techniques

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

Tina Huang, Gunnar Mein, Jeff Li

8/12/2020

Contents

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.

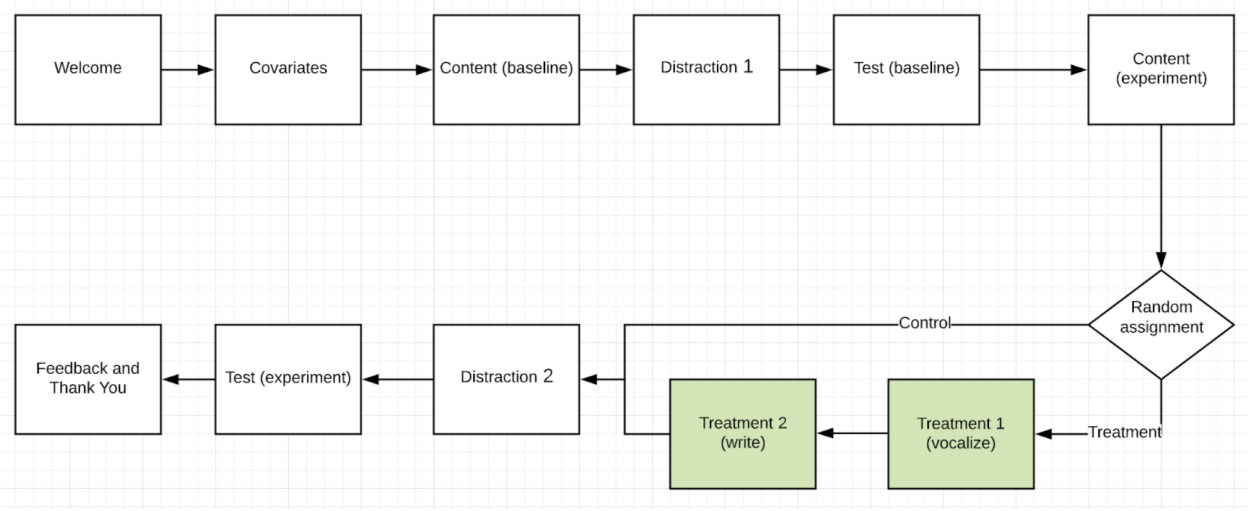
Hypothesis

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.

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.

Treatment

todo: [2 screenshots, one from treatment and one from control]

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.

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 one 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.

Analysis of Results

Loading and Preprocessing the Data

```
dataset = load_data()
```

```
## Warning in ifelse(as.numeric(as.character(rote_cov$age)) > 100, NA,
## as.numeric(as.character(rote_cov$age))): NAs introduced by coercion

## Warning in ifelse(as.numeric(as.character(rote_cov$age)) > 100, NA,
## as.numeric(as.character(rote_cov$age))): NAs introduced by coercion
```

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

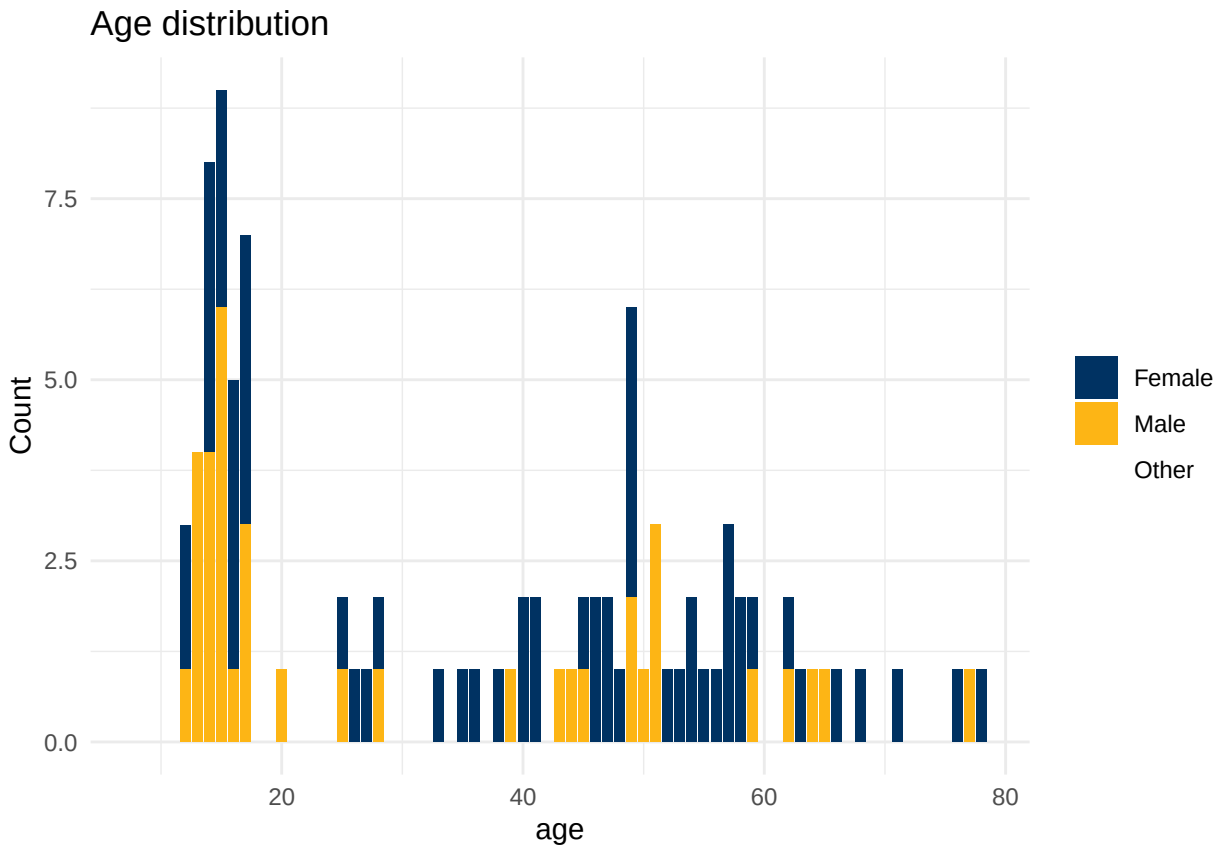
EDA

```
stargazer(dataset, header=FALSE)
```

Table 1:

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
score	97	0.979	1.671	−2	0	2	5
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

```
plot_age()
```



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 be 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.

```
stargazer(cov_check_regression(), header=FALSE)
```

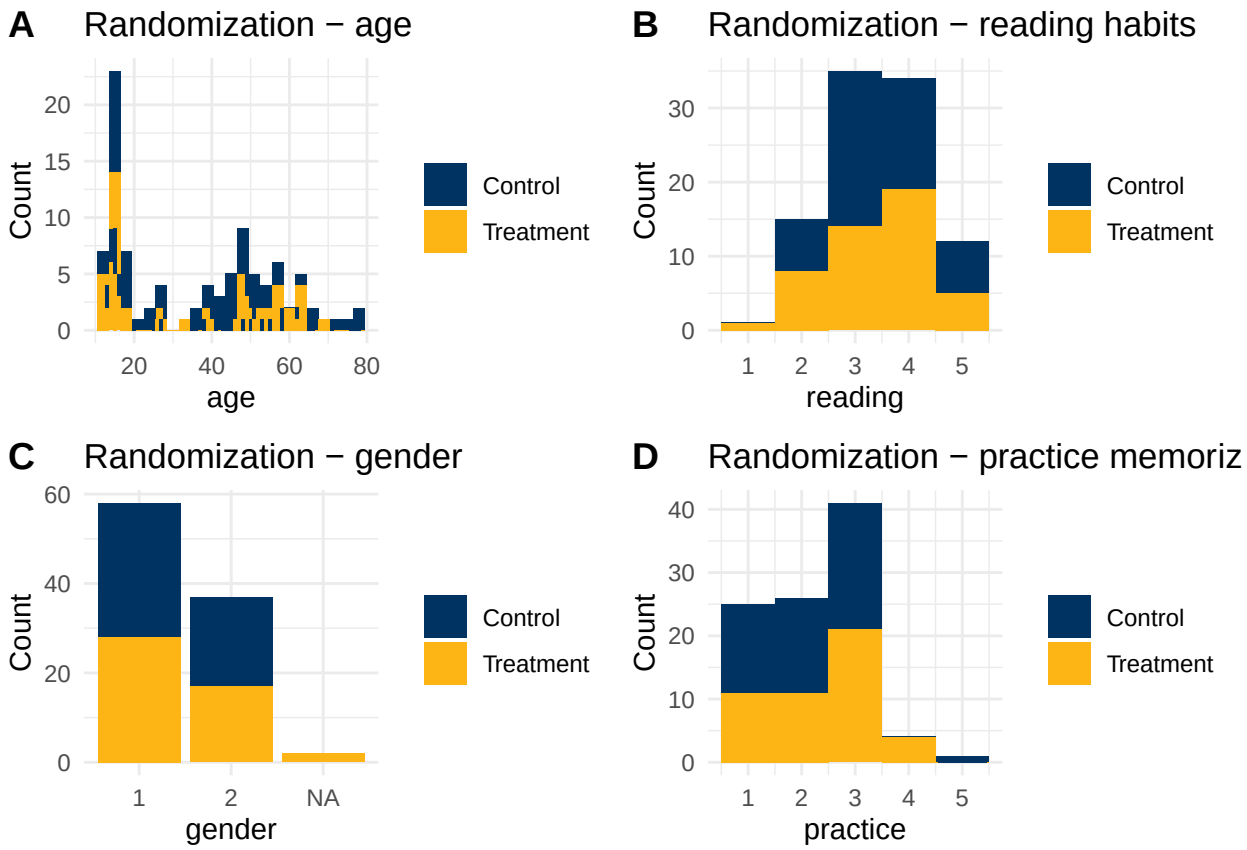
Visual Check

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.

```
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)
```

Table 2:

	Dependent variable:
	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
Note:	*p<0.1; **p<0.05; ***p<0.01



Effect Calculation

todo: everything

```
result_regressions()
```

Table 3: Regression Results with Clustered Standard Errors

	Dependent variable:	
	score	
	(1)	(2)
treat	0.659*	0.663*
	(0.319)	(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**	0.126
	(0.245)	(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)

Appendix

Code

```
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
}


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,
```

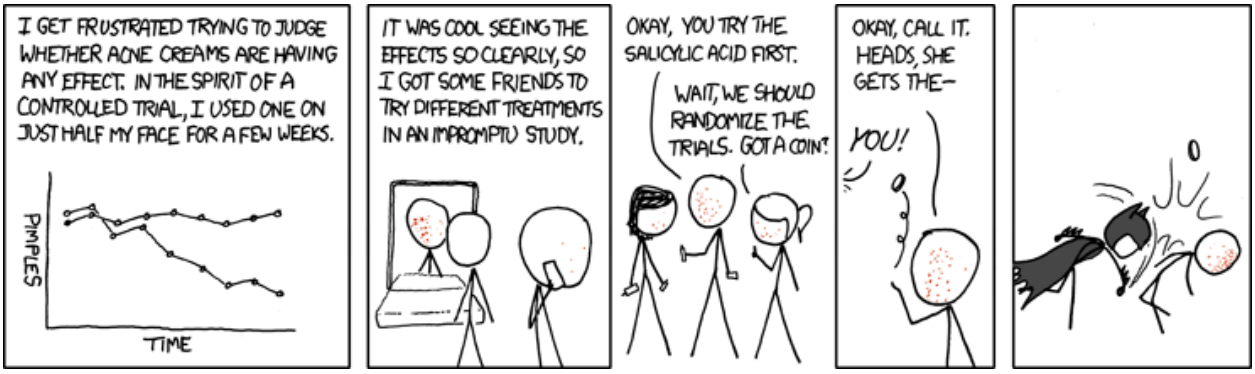
```
            type = "latex",
            omit.table.layout= "n",
            keep.stat = c("adj.rsq", "n", "f", "ser", "aic", "wald"),
            se = list(sqrt(diag(clustered_errors1)),sqrt(diag(clustered_errors2))),
            star.cutoffs = c(0.05, 0.01, 0.001),
            title="Regression 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=treat))+
    geom_density(stat="count", alpha=0.3)+
    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_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_brewer(palette="Dark2") +
    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())
}
```



https://xkcd.com/700/