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

Experiment Design

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, Regression modeling

Randomization Checks

Effect Calculation

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
}


calc_regressions = function() {
  regression1 <- lm(score ~ treat + age + prior_knowledge + reading + gender + practice,data=dataset)
  regression2 <- lm(score ~ treat ,data=dataset)
  clustered_errors1 <- vcovCL_1c <- vcovCL(regression1, cluster = dataset[ , 'cluster'])
  clustered_errors2 <- vcovCL_2c <- vcovCL(regression2, cluster = dataset[ , 'cluster'])
}

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

```
print_regressions = function() {
  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")
}
```

```
plot_rand_gender = function (){
  dataset %>% ggplot(aes(x=gender, fill=treat))+
    geom_bar(stat="count", 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=treat))+
    geom_bar(stat="count", position="stack", binwidth=3) +
    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=treat))+
    geom_bar(stat="count", position="stack", binwidth=1) +
    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=treat))+
    geom_bar(stat="count", position="stack", binwidth=1) +
    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_bar(stat="count", position="dodge") +
    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=gender)) +
  geom_bar() +
  ggtitle("Distribution of Gender")
}

plot_age = function() {
dataset %>% ggplot(aes(x=age, fill=gender))+
    geom_bar(stat="count", position="stack", binwidth=3) +
    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(stat="count", position="dodge") +
    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())
}
```