[corrected]0207_Shuxi Chen_Problem Set 1

2025-02-07

Packages Used:

```
pacman::p_load(tidyverse,gt,foreign,knitr,webshot2,
               reshape2,viridis,ggthemes,stargazer,texreg,
               sandwich,modelsummary,Rmisc,neatStats,gmodels, dplyr, tidyr)
```

Lab experiment

Load and Inspect the Data

```
df_game <- read.csv("incumbentGame.csv")
```

check NA

```
table(is.na(df_game$column_name))
```

```
## < table of extent 0 >
```

1. The estimated expectation

• What is the estimated expectation of kept conditional on type and late.luck? Answer using a 2x2 table like the one below.

```
# "kept" means of each combination of "type" and "late.luck"
mean_table <- tapply(df_game$kept, list(df_game$type, df_game$late.luck), mean, na.rm = TRUE)

# reshape to 2x2
result_table <- as.data.frame.matrix(mean_table)

colnames(result_table) <- c("Not Lucky", "Lucky")
rownames(result_table) <- c("Low", "High")

result_table
```

```
##      Not Lucky     Lucky
## Low   0.562500 0.6000000
## High  0.747619 0.8119266
```

2. OLS

• You try showing the result to your friends, but they complain that two-way tables aren't fancy enough for social scientists. Instead, regress kept on the other two variables (and a constant term) using OLS and report the results in a table of coefficients and standard errors.

```
library(sandwich)
```

```r
reg <- lm(kept ~ late.luck + type, data = df_game)

# HC2 robust SE
hc2_vcov <- vcovHC(reg, type = "HC2")
robust_se <- sqrt(diag(hc2_vcov))

# coefficient
coefficient <- summary(reg)$coef[, "Estimate"]

# making them into a table
result_table <- data.frame(
  Coefficient = coefficient,
  HC2_SE = robust_se,
  row.names = rownames(summary(reg)$coef)
)

result_table
```

```
##               Coefficient     HC2_SE
## (Intercept)    0.55755840 0.04221262
## late.luckTRUE  0.06089025 0.03051875
## typeTRUE       0.19093916 0.04421944
```

```r
library(estimatr)
lm_robust(kept ~ late.luck + type, data = df_game)
```

```
##                 Estimate Std. Error   t value     Pr(>|t|)    CI Lower
## (Intercept)   0.55755840 0.04221262 13.208333 8.507041e-37 0.474721594
## late.luckTRUE 0.06089025 0.03051875  1.995175 4.629839e-02 0.001001145
## typeTRUE      0.19093916 0.04421944  4.317991 1.733454e-05 0.104164234
##                CI Upper  DF
## (Intercept)   0.6403952 987
## late.luckTRUE 0.1207793 987
## typeTRUE      0.2777141 987
```

3. recover the estimated conditional expectations

• Can you recover the estimated conditional expectations in your first table from the regression output? Explain how.

Yes, as the dummy basis have been set as "Not Lucky (late.luckFALSE)" and "Low (typeFALSE)", so we can extrapolate the expected values for 4 combinations: Low + Not Lucky: β_0=0.5576 Low + Lucky: $\beta_0 + \beta_1$=0.6185 High + Not Lucky: $\beta_0 + \beta_2$=0.7485 High + Lucky: $\beta_0 + \beta_1 + \beta_2$=0.8094

4 model with interaction term

• In what way is your regression specification not as flexible as it could be? Amend the specification in light of your answer. Fit the new model and report the results in a regression table, then briefly describe any new inferences that you can draw.

The interpretation of the coefficients in the model, "holding other variables constant, increasing one unit of A will change the target by X amount", relies on the assumption that late.luck and kept are independent, meaning a change in one does not affect the other. But f the allocator is a high type, the effect of luck might be stronger because high-type allocators are more likely to generate high payments. This suggests that a potential interaction effect could exist, even though participants were unaware of the true types of allocators.

Another limitation is the assumption of linearity in the relationship between variables. In reality, the relationship might be nonlinear. In such cases, using more flexible models like Random Forest or Gradient Boosting could better capture these complexities.

To simplify, I've added an interaction term to the model to increase its flexibility while still maintaining interpretability.

The interaction term is not statistically significant. While the effect of being "Lucky" is smaller compared to the first-order model, the effect of having a "High-type" allocator remains statistically significant in both models. This means that participants may not be as influenced by recent performance as initially hypothesized. Instead, they rely more on the general performance of the allocator than on the luck factor in the last four rounds. And the relationship between "luck" and "kept" appears to be independent of the allocator type.

```
lm_robust(kept ~ late.luck * type, data = df_game)
```

```
##                            Estimate Std. Error    t value     Pr(>|t|)
## (Intercept)             0.56250000 0.04708568 11.9463085 8.097676e-31
## late.luckTRUE           0.03750000 0.10243497  0.3660859 7.143795e-01
## typeTRUE                0.18511905 0.05017008  3.6898298 2.366824e-04
## late.luckTRUE:typeTRUE 0.02680756 0.10722217  0.2500188 8.026249e-01
##                              CI Lower  CI Upper  DF
## (Intercept)              0.47010035 0.6548997 986
## late.luckTRUE           -0.16351560 0.2385156 986
## typeTRUE                 0.08666665 0.2835714 986
## late.luckTRUE:typeTRUE -0.18360232 0.2372174 986
```

Field experiment: Corruption in Indonesia

Load and Inspect the Data

```
df_olken <- read.csv("olken_data.csv")
```

check NA

```
table(is.na(df_olken$column_name))
```

```
## < table of extent 0 >
```

check the distribution of the data

```
summary(df_olken)
```

```
##        id          treat.invite     pct.missing          head.edu
##  Min.   :  1.0   Min.   :0.0000   Min.   :-1.10282   Min.   : 6.00
##  1st Qu.:118.8   1st Qu.:0.0000   1st Qu.: 0.01996   1st Qu.: 9.00
##  Median :236.5   Median :1.0000   Median : 0.22717   Median :12.00
##  Mean   :236.5   Mean   :0.6589   Mean   : 0.23633   Mean   :11.56
##  3rd Qu.:354.2   3rd Qu.:1.0000   3rd Qu.: 0.41690   3rd Qu.:12.00
##  Max.   :472.0   Max.   :1.0000   Max.   : 1.67431   Max.   :20.00
##     mosques          pct.poor        total.budget
##  Min.   :0.0000   Min.   :0.01867   Min.   :  8.758
##  1st Qu.:0.8602   1st Qu.:0.23317   1st Qu.: 53.770
##  Median :1.2629   Median :0.38596   Median : 73.177
##  Mean   :1.4377   Mean   :0.40713   Mean   : 83.229
```

```
##  3rd Qu.:1.8841   3rd Qu.:0.56572   3rd Qu.:103.448
##  Max.   :6.8891   Max.   :0.94462   Max.   :890.242
```

A. Balance table

• Using either base R or the tidyverse1 together with any table package of your choice, create a balance table. Include for each pre-treatment covariate comparisons for treated and untreated units. Report the mean and standard deviation for each covariate within each group. Also report a test, for each covariate, of the hypothesis that the difference in means between treatment conditions is zero.

step 1: Calculate the mean and SD by treatment status for each covariate

```
vars <- df_olken %>%
  select(head.edu, mosques, pct.poor, total.budget)

# mean
bal.mean <- aggregate(vars,
                      by = list(df_olken$treat.invite),
                      function(x) mean(x, na.rm =T)
                      )

# sd
bal.sd <- aggregate(vars,
                    by = list(df_olken$treat.invite),
                    function(x) sd(x, na.rm =T)
                    )

# t-test
diff_means_pval <- function(x) {t.test(
  vars[df_olken$treat.invite == 1, x],
  vars[df_olken$treat.invite == 0, x])$p.value
  }

# loop through each column index and calculates the p-value
p_values = sapply(1:length(vars), diff_means_pval)
```

step 2: balance table

```
# Create an vector to store the difference in means
diff.means <- vector()

for (i in 1:4) {
 diff.means[i] <-
   mean(vars[df_olken$treat.invite == 1, i], na.rm = T) -
 mean(vars[df_olken$treat.invite == 0, i], na.rm = T)
}

# putting the means, sds, differences, and p values all together
bal <- rbind(bal.mean, bal.sd, c(NA, diff.means), c(NA, p_values))

# Transpose the matrix and label the balance table
bal = t(bal)
bal = bal[-1, 1:6]
```

```r
colnames(bal) = c("Control_Mean", "Treat_Mean", "Control_SD",
                  "Treat_SD", "Diff_Means", "ttest_p-val")

balance_table <- as_tibble(bal)

# adding rownames
balance_table <- as.data.frame(balance_table)

# This looks a lot better just by using kable!
balance_table %>%
 mutate(Covariate = row.names(bal)) %>%
 dplyr::select(Covariate, everything()) %>%
 kable(type = "text")
```

Covariate	Control_Mean	Treat_Mean	Control_SD	Treat_SD	Diff_Means	ttest_p-val
head.edu	11.583851	11.5466238	2.7215781	2.7197450	-0.0372271	0.8880239
mosques	1.472887	1.4195176	0.8256950	0.8377566	-0.0533699	0.5081709
pct.poor	0.400366	0.4106243	0.2124669	0.2130463	0.0102584	0.6196482
total.budget	83.354209	83.1643437	42.9089588	60.4109790	-0.1898659	0.9685551

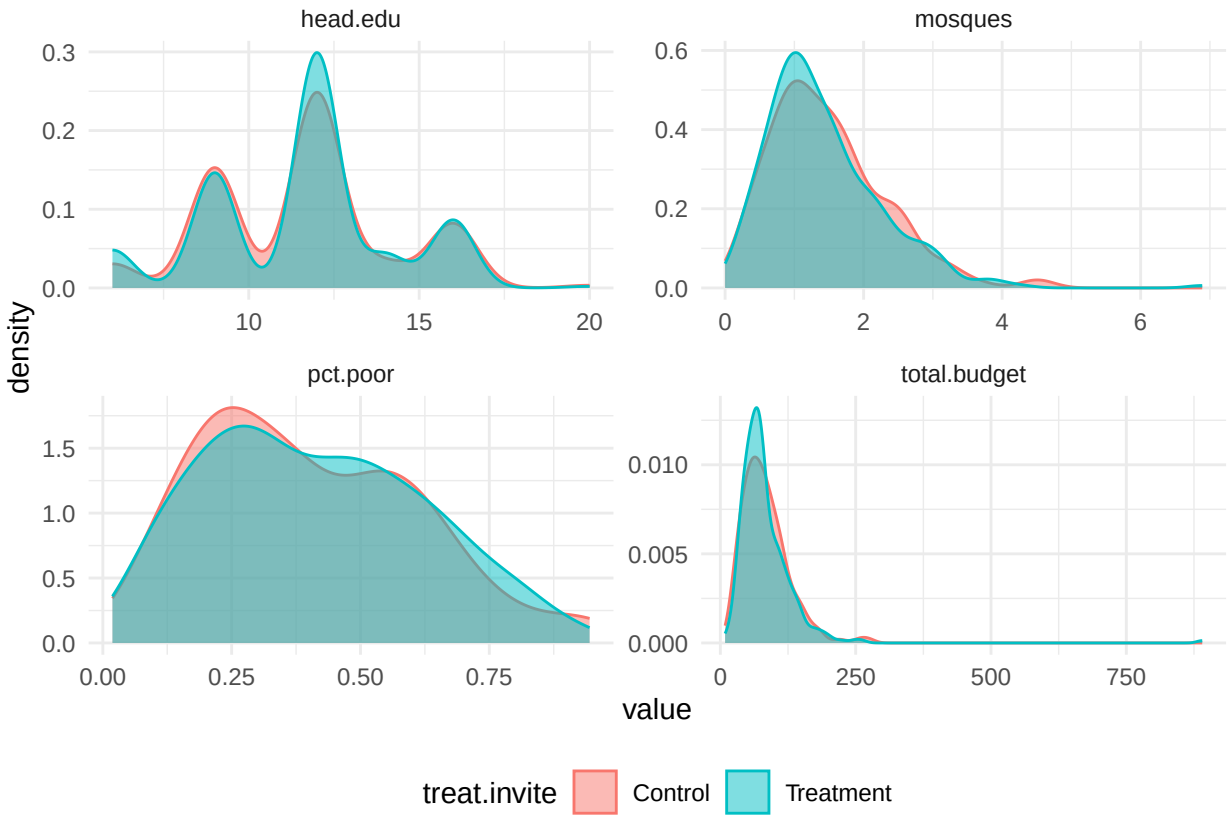
2. Visualizing the Distrubitions of Covariates

• For each covariate, plot its distributions under treatment and control (side by side or overlaying). Include the plots in your write-up.

```r
# prepare the data
data_plot <- df_olken %>%
  select(head.edu, mosques, pct.poor, total.budget, treat.invite) %>%
  dplyr::mutate(id = 1:dplyr::n(),
         treat.invite = ifelse(treat.invite == 1,"Treatment", "Control")
         )

# use pivot longer for easy plotting with facets in ggplot
data.pivot <- data_plot %>%
  pivot_longer(cols = 1:4, names_to = "variable")

# plot the densities
ggplot(data.pivot, aes(x = value, fill = treat.invite,
 color = treat.invite)) +
 geom_density(alpha = 0.5) +
 facet_wrap(~ variable, scales = "free") +
 theme_minimal() +
 theme(legend.position = "bottom") # Reposition the legend to the bottom of the figure
```



3. the importance of checking balance

• With reference to your table and plots, do villages in each condition appear similar in the pre-treatment covariates? Explain the importance of checking balance in a randomized experiment and the result you typically expect to find.

Yes, the mean differences between the treatment and control groups for all covariates are very small and statistically insignificant (p-value > 0.05). Plus, the distributions of all covariates are nearly identical, indicating no systematic differences in pre-treatment characteristics between the two groups. This shows that randomization was successful, achieving balance and reducing the risk of bias. So we can more confidently attribute any observed differences in the outcome to the treatment effect.

4. F-Statistic

• Regress treatment on the pre-treatment covariates and report the p-value of an omnibus F-test. What do you conclude from the results?

The p-value of F-test is very high, meaning the null hypothesis is not rejected ($H0 : \beta_0 = \beta_1 = \beta_2 = \beta_3 = \beta_4 = 0$). This confirms that the randomization process successfully ensured balance across these covariates. It validates the earlier statement that any observed differences between the groups can be attributed to the intervention rather than pre-existing disparities.

```
reg <- lm(treat.invite ~ head.edu + mosques + pct.poor + total.budget, data = df_olken)
summary(reg)
```

```
##
## Call:
## lm(formula = treat.invite ~ head.edu + mosques + pct.poor + total.budget,
##     data = df_olken)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7075 -0.6474  0.3263  0.3449  0.4480
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.767e-01  1.172e-01   5.772 1.43e-08 ***
## head.edu     -8.972e-04  8.099e-03  -0.111    0.912
## mosques      -1.871e-02  2.646e-02  -0.707    0.480
## pct.poor      5.744e-02  1.044e-01   0.550    0.582
## total.budget -4.698e-05  4.013e-04  -0.117    0.907
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4762 on 467 degrees of freedom
## Multiple R-squared:  0.00163,    Adjusted R-squared:  -0.006921
## F-statistic: 0.1906 on 4 and 467 DF,  p-value: 0.9433
```

B. Treatment effects

1. calculate ATE

• Use the difference-in-means estimator to estimate the average treatment effect and its standard error, i.e., do not use OLS to estimate the average treatment effect.

```
# Apply the difference in means estimator
values <- df_olken %>% group_by(treat.invite) |>
  summarize(jobs = mean(pct.missing, na.rm = T))

ybar <- tapply(df_olken$pct.missing,
 list('treated'= df_olken$treat.invite),
 function(x) mean(x, na.rm = T)
 )

ybar['1'] - ybar['0']
```

```
##           1
## -0.02494953
```

```
# Estimate the standard error of the difference in means
df_olken_2 <- df_olken %>%
  select(pct.missing, treat.invite) %>%
  drop_na()

seDiffMeans <- function(y, tx){
 y1 = y[tx == 1]
 y0 = y[tx == 0]
 n1 = length(y1) # Number of observations in the  treatment group
 n0 = length(y0) # Number of observations in the control group

 sqrt(((var(y1) / n1 + var(y0) / n0)))
}

seDiffMeans(df_olken_2$pct.missing, df_olken_2$treat.invite)
```

```
## [1] 0.03310019
```

2. Computing ATE with Bivariate OLS

• Now estimate the average treatment effect and its standard error using a bivariate regression of outcomes on treatment. Are the results different from before? If so, why? Make the changes necessary for them to match exactly, and explain your method.

The coefficient on treat.invite in the regression is identical to the ATE obtained from the difference-in-means estimator in a bivariate regression. This is because the treatment is binary, making the coefficient mathematically equal to the difference in means.

But the standard errors differ, as the SE from the regression assume homoskedasticity, meaning the variance of the outcome variable is constant within the treatment and control groups. In contrast, the standard error from the seDiffMeans function accounts for potential heteroskedasticity, providing a more reliable measure of uncertainty when variances differ across groups.

```
mod.bivariate <- lm(pct.missing ~ treat.invite, data = df_olken)
summary(mod.bivariate)
```

```
##
## Call:
## lm(formula = pct.missing ~ treat.invite, data = df_olken)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.33064 -0.21249 -0.01284  0.18281  1.42154
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.25277    0.02716   9.306   <2e-16 ***
## treat.invite -0.02495    0.03346  -0.746    0.456
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3447 on 470 degrees of freedom
## Multiple R-squared:  0.001181,   Adjusted R-squared:  -0.0009438
## F-statistic: 0.5559 on 1 and 470 DF,  p-value: 0.4563
```

To match them, we can use vcovHC in the sandwich package, setting the type = "HC2" to calculate the robust SE. The code would look like this:

```
mod.bivariate <- lm(pct.missing ~ treat.invite, data = df_olken)

# robust estimate
se.bivariate <- sqrt(
  diag(
    vcovHC(
      mod.bivariate, type = 'HC2')))

options(scipen = 999) # control the use of scientific notation in numeric output

stargazer(mod.bivariate,
          se = list(se.bivariate),
          digits = 8,
          notes = "HC2 Robust SEs",
          type = "text")
```

```
##
```

```
## ================================================
##                          Dependent variable:
##                      ---------------------------
##                                pct.missing
## ------------------------------------------------
## treat.invite                  -0.02494953
##                              (0.03310019)
##
## Constant                     0.25276710***
##                              (0.02654878)
##
## ------------------------------------------------
## Observations                      472
## R2                             0.00118133
## Adjusted R2                    -0.00094382
## Residual Std. Error    0.34466200 (df = 470)
## F Statistic          0.55588150 (df = 1; 470)
## ================================================
## Note:                *p<0.1; **p<0.05; ***p<0.01
##                                    HC2 Robust SEs
```

4. Re-estimation

• Re-estimate the average treatment effect using a regression specification that includes pre-treatment covariates (additively and linearly). Report your estimates of the treatment effect and its standard error. Do you expect them to differ from the difference-in-means estimates, and do they? If so, why?

Here I used a robust regression model, which employs HC2 standard errors by default when running OLS. As expected, the treatment difference in means aligns with the OLS results, assuming other covariates remain constant.

```
reg_covariates <- lm_robust(pct.missing ~ treat.invite + head.edu + mosques +
                      pct.poor + total.budget, data = df_olken)
summary(reg_covariates)
```

```
##
## Call:
## lm_robust(formula = pct.missing ~ treat.invite + head.edu + mosques +
##     pct.poor + total.budget, data = df_olken)
##
## Standard error type:  HC2
##
## Coefficients:
##                Estimate Std. Error t value   Pr(>|t|)   CI Lower  CI Upper  DF
## (Intercept)   0.3904455  0.0937358  4.1654 0.00003705  0.2062483  0.574643 466
## treat.invite -0.0264183  0.0326312 -0.8096 0.41858335 -0.0905409  0.037704 466
## head.edu     -0.0055082  0.0061452 -0.8963 0.37053332 -0.0175838  0.006567 466
## mosques      -0.0481914  0.0184408 -2.6133 0.00925700 -0.0844288 -0.011954 466
## pct.poor     -0.1177125  0.0733673 -1.6044 0.10929721 -0.2618842  0.026459 466
## total.budget  0.0005307  0.0003221  1.6475 0.10012015 -0.0001023  0.001164 466
##
## Multiple R-squared:  0.0294 ,    Adjusted R-squared:  0.01898
## F-statistic: 2.626 on 5 and 466 DF,  p-value: 0.02348
```

C. Heterogeneous effects

1. calculate ATE for poverty levels

• Estimate the ATE for villages with more than half of households below the poverty line, and then do the same for villages with less than half of households below the poverty line.

Originally, I used an OLS model to estimate the ATE, where the ATE is represented by β_1. I included villages with a poverty level of 0.5 in the "Poor" group (using >= 0.5). While my result for the "Poor" group matches the DiM-based result, the ATE for the "Rich" group is slightly larger than the DiM-based ATE. This is because β_1 (Sxx/Sxy) is a conditional-variance-weighted ATE, where larger deviations from the mean have a greater influence on the coefficient estimates. This implicitly assigns weights to observations based on their influence on the model. In contrast, DiM calculates a simple unweighted average of treatment effects. **Note: Only if the treatment effect is constant across subgroups can we interpret β_1 as the ATE or ATT; otherwise, it's a conditional-variance-weighted ATE**

```
# > 50
over_50 <- df_olken[df_olken$pct.poor > 0.5,]
reg_over_50 <- lm(pct.missing ~ treat.invite, data = over_50)
summary(reg_over_50)
```

```
##
## Call:
## lm(formula = pct.missing ~ treat.invite, data = over_50)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.28590 -0.19345  0.02717  0.16049  1.35337
##
## Coefficients:
##              Estimate Std. Error t value    Pr(>|t|)
## (Intercept)   0.25696    0.04627   5.554 0.000000115 ***
## treat.invite -0.07387    0.05702  -1.296       0.197
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3431 on 159 degrees of freedom
## Multiple R-squared:  0.01045,    Adjusted R-squared:  0.004224
## F-statistic: 1.679 on 1 and 159 DF,  p-value: 0.197
```

```
print("--------------------------------")
```

```
## [1] "--------------------------------"
```

```
# < 50
less_50 <- df_olken[df_olken$pct.poor < 0.5,]
reg_less_50 <- lm(pct.missing ~ treat.invite, data = less_50)
summary(reg_less_50)
```

```
##
## Call:
## lm(formula = pct.missing ~ treat.invite, data = less_50)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.23283 -0.22572 -0.02433  0.18494  1.42371
##
## Coefficients:
```

```
##              Estimate Std. Error t value          Pr(>|t|)
## (Intercept)  0.250594   0.033616   7.455 0.000000000000926 ***
## treat.invite 0.000692   0.041474   0.017             0.987
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3461 on 307 degrees of freedom
## Multiple R-squared:  9.068e-07,  Adjusted R-squared:  -0.003256
## F-statistic: 0.0002784 on 1 and 307 DF,  p-value: 0.9867
```

DiM methods

```
df_olken <- df_olken %>%
  mutate(Wealth = ifelse(pct.poor > 0.5, "Poor", "Rich"))

mean_poor_treated <- mean(df_olken$pct.missing[df_olken$Wealth == "Poor" & df_olken$treat.invite == 1],
mean_poor_control <- mean(df_olken$pct.missing[df_olken$Wealth == "Poor" & df_olken$treat.invite == 0],
ATE_poor <- mean_poor_treated - mean_poor_control

mean_rich_treated <- mean(df_olken$pct.missing[df_olken$Wealth == "Rich" & df_olken$treat.invite == 1],
mean_rich_control <- mean(df_olken$pct.missing[df_olken$Wealth == "Rich" & df_olken$treat.invite == 0],
ATE_rich <- mean_rich_treated - mean_rich_control


print(paste("ATE for Poor:", ATE_poor))
```

```
## [1] "ATE for Poor: -0.0738740968341338"
```

```
print(paste("ATE for Rich:", ATE_rich))
```

```
## [1] "ATE for Rich: 0.000355573304049728"
```

2. test the null hypothesis

- Estimate the standard error of the difference in treatment effects and test the null hypothesis that there is no difference between them. What do you conclude?

OLS methods The 95% CI doesn't contain 0, indicating no significant difference in the treatment effects between villages with higher and lower poverty levels. This suggests that poverty level may not be a strong covariate influencing the intervention's outcome.

```
# means
mean_over_50 <- coef(reg_over_50)["treat.invite"]
mean_less_50 <- coef(reg_less_50)["treat.invite"]

diff_mean <- mean_over_50 - mean_less_50

# robust SEs
se_over_50 <- sqrt(diag(vcovHC(reg_over_50, type = "HC2")))["treat.invite"]
se_less_50 <- sqrt(diag(vcovHC(reg_less_50, type = "HC2")))["treat.invite"]

se_both <- sqrt(se_over_50^2 + se_less_50^2)

# t-test
t_result <- diff_mean / se_both

# 95% CI
```

```r
ci_lower <- diff_mean - 1.96 * se_both
ci_upper <- diff_mean + 1.96 * se_both
paste0(round(ci_lower, 4), ", ", round(ci_upper, 4))
```

```
## [1] "-0.2082, 0.0591"
```

DiM methods

```r
# mean
diff_mean <- ATE_poor - ATE_rich

se_poor <- seDiffMeans(df_olken$pct.missing[df_olken$pct.poor > .5], df_olken$treat.invite[df_olken$pct
se_rich <- seDiffMeans(df_olken$pct.missing[!df_olken$pct.poor > .5], df_olken$treat.invite[!df_olken$p

se_both <- sqrt(se_poor^2 + se_rich^2)
# t-test
t_result <- diff_mean / se_both

# 95% CI
ci_lower <- diff_mean - 1.96 * se_both
ci_upper <- diff_mean + 1.96 * se_both
paste0(round(ci_lower, 4), ", ", round(ci_upper, 4))
```

```
## [1] "-0.2077, 0.0593"
```