Econometrics II - Assignment 3

Floris Holstege, Stanislav Avdeev

```
# create df with treatment effects
dfTreatment <- data.frame(N_treated = c(100, 75, 25), N_control = c(100, 25, 75), Avg_Outcome_treated =
row.names(dfTreatment) <- c("Purple", "Blue", "Green")

# average treatment effect per group
ATE_group <- dfTreatment$Avg_Outcome_treated - dfTreatment$Avg_Outcome_control

ATE_treated <- sum(dfTreatment$N_treated/sum(dfTreatment$N_treated) * dfTreatment$Avg_Outcome_treated)
ATE_control <- sum(dfTreatment$N_control/sum(dfTreatment$N_control) * dfTreatment$Avg_Outcome_control)
ATE_treated - ATE_control
```

Problem 1

Before we answer this question, let's lay out some basic terms that are useful for understanding our subsequent. We define the average treatment effect (ATE) as:

$$ATE = \mathbb{E}(\delta) = \mathbb{E}(Y_1^* - Y_0^*) = \mathbb{E}(Y_1^*) - \mathbb{E}(Y_0^*).$$

Where Y_1^* is the latent variable of interest for the group that received treatment. For this group, $D_i = 1$. Y_0^* is the latent variable of interest for the group that did not receive treatment. We only observe $\mathbb{E}(Y_1^*)$, since our control group does not receive the treatment. Given we only observe the effect when $D_i = 1$, we define the average treatment effect of the treated as (ATET).

$$ATET = \mathbb{E}(\delta|D = 1) = \mathbb{E}(Y_1^* - Y_0^*|D = 1) = \mathbb{E}(Y_1^*|D = 1) - \mathbb{E}(Y_0^*|D = 1).$$

If the treatment is assigned randomly, then $(Y_{1i}^*, Y_{0i}^*) \perp D_i$. In this case, ATE = ATET, since there is no significant difference between the characteristics of the treatment and control group. For this question, we assume that the treatment has been randomly assigned, and thus that ATE = ATET.

- Average treatment per group: $ATE_{purple} = 9 - 7 = 2$, $ATE_{blue} = 13 - 8 = 5$, $ATE_{green} = 10 - 9 = 1$
- Average treatment for the full population: $ATE = \mathbb{E}(Y_1^*) - \mathbb{E}(Y_0^*) = 10.625 - 7.875 = 2.75$
- Average treatment for the treated: $ATE = ATET = 2.75$

Problem 2

I)

```
dfBonus <- read.dta("Data/bonus.dta")

dfBonus$category <- ifelse(dfBonus$bonus500 == 1, "Low-reward", ifelse(dfBonus$bonus1500 == 1, "High-rew

dfSummaryStats <- dfBonus %>%
                              group_by(category) %>%
                              summarise(perc_passed_year1 = sum(pass)/ n(),
                                        avg_myeduc = mean(myeduc),
                                        avg_fyeduc = mean(myeduc),
                                        avg_p0 = mean(p0),
                                        math = mean(math[!is.na(math)]),
                                        perc_job = sum(job[!is.na(job)])/n(),
                                        avg_effort = mean(effort[!is.na(effort)]))

# COMMENT FLO: How to deal with NA's? I think we need to just mention this as a limitation in checking l
```

II)

III)

```
# define the three LPM models
LPM_simple <- lm(pass ~ category, data=dfBonus)
LPM_addedRegressors <- lm(pass ~ category + math + fyeduc + p0, data=dfBonus)
LPM_allRegressors <- lm(pass ~ category + math + fyeduc + p0 + effort + job, data=dfBonus)

# check adjusted R2 and coefficients
# COMMENT FLO: There does not seem to be an effect of the treatment
summary(LPM_simple)
summary(LPM_addedRegressors)
summary(LPM_allRegressors)


# COMMENT FLO: my preferred model is the one with the most variables - 1) All of these capture variables
mX <- dfBonus %>% select(math, fyeduc, p0, effort, job) %>% as.matrix() %>% na.omit()
cor(mX)
vif(LPM_allRegressors)
```

IV)

```
LPM_drop <- lm(dropout ~ category + math + fyeduc + p0 + effort + job, data=dfBonus)
LM_pointsYear1 <- lm(stp2001 ~ category + math + fyeduc + p0 + effort + job, data=dfBonus)
LM_pointsYear3 <- lm(stp2004 ~ category + math + fyeduc + p0 + effort + job, data=dfBonus)

summary(LPM_drop)
summary(LM_pointsYear1)
summary(LM_pointsYear3)

# COMMENT FLO: Took my preferred model here - can obviously change but seems like it again has no effec
```

V)

VI)

```
n <- nrow(dfBonus)
p <-
```