Econometrics II - Assignment 3

Uncensored sloths

20 Jan 2022

Question 1

(i) Compute for each of the colors the treatment effect.

The average treatment effects of each color can be estimated by taking the difference between the average outcome of treated and the average outcome of the control group. Hence, the average treatment effect of purple is 2, the average treatment effect of blue is 5 and the average treatment effect of green is 1.

(ii) Compute the average treatment effect in the full population.

To compute the average treatment, we have to compute the overall average of the treated group and the overall average of the control group and take the difference between both of them:

```
(100*9+75*13+25*10)/200 - (7*100+8*25+9*75)/200
```

```
## [1] 2.75
```

(iii) Compute the average treatment effect on the treated.

As we do not have any information on the selection process, we assume that the selection process was random. When the selection process is randomized and therefore the treatment assignment is statistically independent of potential outcomes, we have:

$$E[Y_1^*] = E[Y_1^*|D = 1] = E[Y_1^*|D = 0] \text{ and } E[Y_0^*] = E[Y_0^*|D = 1] = E[Y_0^*|D = 0]$$

Hence, the average treatment effect on the treated corresponds to the average treatment effect, $ATE = ATET = 2.75$.

(iv) Give an example where the average treatment effect on the treated would be more useful to consider than the average treatment effect, and explain why.

The average treatment effect on the treated is more applicable when we want to now the effect of a treatment on the treated group. For example, when conduction cost-benefit analysis of a treatment, we are interested to know whether the effect is sufficient given the costs such as in job support programs where we want to know whether the support measures helped the participants to gain a job. Another situation is the effect of medication where we are interested to know whether a certain medicine helped the participants or not.

Question 2

```
# Load data
data <- read.csv("assignment3.csv")

data$important[data$important == "sport"] <- 0
data$important[data$important == "school"] <- 1
data$better[data$better == "Math"] <- 1
```

```r
data$better[data$better == "Language"] <- 0
data$preferredhand[data$preferredhand == "Left"] <- 1
data$preferredhand[data$preferredhand == "Right"] <- 0
```

```r
data$treatment [data$treatment  == "Nothing"] <- 0
data$treatment [data$treatment  == "Candybar"] <- 2
data$treatment [data$treatment  == "Fruit"] <- 1

data$important[data$important == ""] <- NA

data %<>%
      mutate_each(funs(if(is.character(.)) as.integer(.) else .))
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
## Warning: `mutate_each_()` was deprecated in dplyr 0.7.0.
## Please use `across()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

(i) Compute the fraction of pupils in all three groups (control, fruit, and candy bar) that have the number correct and that are expected to lie. Show within a table whether pupil characteristics are balanced over the treatment groups.

```r
balance <- balance_table(data[, !names(data) %in% "id"], "treatment")
balance
```

```
## # A tibble: 9 x 6
##   variables1    Media_control1 Media_trat1 Media_trat2 p_value1 p_value2
##   <chr>                  <dbl>       <dbl>       <dbl>    <dbl>    <dbl>
## 1 better                 0.511       0.625       0.471   0.173   0.648
## 2 correct                0.398       0.362       0.412   0.662   0.872
## 3 gender                 0.277       0.293       0.392   0.829   0.168
## 4 grade                  3.03        3.5         3.10    0.0299  0.807
## 5 important              0.870       0.845       0.667   0.679   0.00873
## 6 oftenexpelled          0.138       0.109       0.137   0.600   0.986
## 7 preferredhand          0.138       0.121       0.157   0.754   0.768
## 8 siblings               0.989       0.914       0.980   0.0549  0.689
## 9 youngestchild          0.351       0.362       0.431   0.892   0.351
```

For all three groups we can see that the average of correct numbers is above $\frac{1}{6}$ which is why calculate the fraction of pupils that lie using the given formula $\frac{x-\frac{1}{6}}{1-\frac{1}{6}}$:

```r
fraction_nothing <- (balance[2, 2]-(1/6))/(1- (1/6))
fraction_fruit <- (balance[2, 3]-(1/6))/(1- (1/6))
fraction_candybar <-(balance[2, 4]-(1/6))/(1- (1/6))
fractions <- cbind(fraction_nothing, fraction_fruit, fraction_candybar)
fractions
```

```
##   Media_control1 Media_trat1 Media_trat2
## 1      0.2774194   0.2344828   0.2941176
```

```r
data$treatment_fruit [data$treatment  == 0] <- 0
data$treatment_fruit [data$treatment  == 2] <- 0
data$treatment_fruit [data$treatment  == 1] <- 1
data$treatment_candy [data$treatment  == 0] <- 0
data$treatment_candy [data$treatment  == 2] <- 1
data$treatment_candy [data$treatment  == 1] <- 0
```

As the table shows, the characteristics are not fully balanced at important, grade and siblings (but not at a high significance level.)

(ii) Use the linear probability model to regress the dummy variable for having the correct number on the dice on the assignment to the three treatment groups. Do you detect significant lying? Show how robust you results are to including additional covariates.

```r
model1 <- lm(correct ~ treatment_fruit + treatment_candy, data = data)
model1_robust <- coeftest(model1, vcov = vcovHC, type = "HC1")
model1_robust
```

```
##
## t test of coefficients:
##
##                  Estimate Std. Error t value  Pr(>|t|)
## (Intercept)      0.397849   0.051135  7.7804 3.844e-13 ***
## treatment_fruit -0.035780   0.081591 -0.4385    0.6615
## treatment_candy  0.013915   0.086231  0.1614    0.8720
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

they are not significantly different from α

```r
lh_robust(correct ~ treatment_fruit + treatment_candy, data = data, linear_hypothesis =
            "(Intercept) = 0.167")
```

```
## Loading required namespace: car

## $lm_robust
##                    Estimate Std. Error    t value     Pr(>|t|)   CI Lower
## (Intercept)      0.39784946 0.05102911  7.7965194 3.485201e-13  0.2972223
## treatment_fruit -0.03578050 0.08158532 -0.4385654 6.614522e-01 -0.1966632
## treatment_candy  0.01391524 0.08630331  0.1612365 8.720707e-01 -0.1562711
##                  CI Upper  DF
## (Intercept)     0.4984767 199
## treatment_fruit 0.1251022 199
## treatment_candy 0.1841016 199
##
## $lh
##                     Estimate Std. Error t value  Pr(>|t|) CI Lower CI Upper  DF
## (Intercept) = 0.167   0.2308    0.05103   4.524 1.043e-05   0.1302   0.3315 199
```

```r
lh_robust(correct ~ treatment_fruit + treatment_candy, data = data, linear_hypothesis =
            "treatment_fruit = 0")
```

```
## $lm_robust
##                    Estimate Std. Error    t value     Pr(>|t|)   CI Lower
## (Intercept)      0.39784946 0.05102911  7.7965194 3.485201e-13  0.2972223
## treatment_fruit -0.03578050 0.08158532 -0.4385654 6.614522e-01 -0.1966632
## treatment_candy  0.01391524 0.08630331  0.1612365 8.720707e-01 -0.1562711
##                  CI Upper  DF
## (Intercept)     0.4984767 199
## treatment_fruit 0.1251022 199
## treatment_candy 0.1841016 199
##
## $lh
##                     Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper  DF
## treatment_fruit = 0 -0.03578    0.08159 -0.4386   0.6615  -0.1967   0.1251 199
```

```r
lh_robust(correct ~ treatment_fruit + treatment_candy, data = data, linear_hypothesis =
            "treatment_candy = 0")
```

```
## $lm_robust
##                    Estimate Std. Error    t value     Pr(>|t|)   CI Lower
## (Intercept)      0.39784946 0.05102911  7.7965194 3.485201e-13  0.2972223
## treatment_fruit -0.03578050 0.08158532 -0.4385654 6.614522e-01 -0.1966632
## treatment_candy  0.01391524 0.08630331  0.1612365 8.720707e-01 -0.1562711
##                  CI Upper  DF
## (Intercept)     0.4984767 199
## treatment_fruit 0.1251022 199
## treatment_candy 0.1841016 199
##
## $lh
##                     Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper  DF
## treatment_candy = 0  0.01392     0.0863  0.1612   0.8721  -0.1563   0.1841 199
```

```r
t.test(subset(data, treatment == 0)$correct, y = NULL,
       alternative = c("greater"),
       mu = 1/6, paired = FALSE, var.equal = FALSE,
       conf.level = 0.95)
```

```
##
##  One Sample t-test
##
## data:  subset(data, treatment == 0)$correct
## t = 4.5304, df = 92, p-value = 8.82e-06
## alternative hypothesis: true mean is greater than 0.1666667
## 95 percent confidence interval:
##  0.3130602       Inf
## sample estimates:
## mean of x
## 0.3978495
```

```r
t.test(subset(data, treatment == 1)$correct, y = NULL,
       alternative = c("greater"),
       mu = 1/6, paired = FALSE, var.equal = FALSE,
       conf.level = 0.95)
```

```
##
##  One Sample t-test
##
## data:  subset(data, treatment == 1)$correct
## t = 3.0696, df = 57, p-value = 0.00164
## alternative hypothesis: true mean is greater than 0.1666667
## 95 percent confidence interval:
##  0.2556329       Inf
## sample estimates:
## mean of x
##  0.362069
```

```
t.test(subset(data, treatment == 2)$correct, y = NULL,
       alternative = c("greater"),
       mu = 1/6, paired = FALSE, var.equal = FALSE,
       conf.level = 0.95)
```

```
##
##  One Sample t-test
##
## data:  subset(data, treatment == 2)$correct
## t = 3.5215, df = 50, p-value = 0.0004631
## alternative hypothesis: true mean is greater than 0.1666667
## 95 percent confidence interval:
##  0.2951201       Inf
## sample estimates:
## mean of x
## 0.4117647
```

```
model2 <- lm(correct ~ treatment_fruit + treatment_candy + gender + grade +  important + better + prefe
model2_robust <- coeftest(model2, vcov = vcovHC, type = "HC1")
model2_robust
```

```
##
## t test of coefficients:
##
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.3509565  0.2204999  1.5916  0.11318
## treatment_fruit -0.0687466  0.0832231 -0.8261  0.40984
## treatment_candy  0.0478029  0.0905649  0.5278  0.59825
## gender          -0.1783695  0.0756028 -2.3593  0.01936 *
## grade           -0.0191890  0.0256185 -0.7490  0.45480
## important        0.1171151  0.0861831  1.3589  0.17584
## better          -0.0029414  0.0722615 -0.0407  0.96757
## preferredhand   -0.0251078  0.1041144 -0.2412  0.80970
## siblings         0.0659319  0.1691014  0.3899  0.69706
## youngestchild    0.0490706  0.0783999  0.6259  0.53216
## oftenexpelled   -0.1177254  0.1037574 -1.1346  0.25801
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model3 <- lm(correct ~ treatment_fruit + treatment_candy + gender + better + preferredhand + siblings +
model3_robust <- coeftest(model3, vcov = vcovHC, type = "HC1")
model3_robust
```

```
##
```

```
## t test of coefficients:
##
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.369338   0.204074  1.8098  0.07192 .
## treatment_fruit -0.077488   0.083503 -0.9280  0.35462
## treatment_candy  0.030912   0.087464  0.3534  0.72416
## gender          -0.174523   0.074342 -2.3476  0.01994 *
## better           0.010222   0.070949  0.1441  0.88560
## preferredhand   -0.074275   0.100333 -0.7403  0.46005
## siblings         0.076604   0.177453  0.4317  0.66647
## youngestchild    0.061196   0.076291  0.8021  0.42349
## oftenexpelled   -0.107126   0.104731 -1.0229  0.30768
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
model4 <- lm(correct ~ treatment_fruit + treatment_candy + gender, data = data)
model4_robust <- coeftest(model4, vcov = vcovHC, type = "HC1")
model4_robust
```

```
##
## t test of coefficients:
##
##                   Estimate Std. Error t value  Pr(>|t|)
## (Intercept)       0.435216   0.055672  7.8175 3.122e-13 ***
## treatment_fruit -0.033972   0.080734 -0.4208   0.67437
## treatment_candy  0.028963   0.086491  0.3349   0.73808
## gender          -0.133656   0.071935 -1.8580   0.06465 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We estimated a model We tried adding different covariates. However, except for gender no covariate was significant and additionally led to higher standard errors. The same happens if one includes combinations of covariates into the model as shown in the second and third model.

(iii) The researcher is interested in testing if boys and girls behave differently. Write down and estimate a model that can test if boys lie more than girls and whether boys and girls respond differently to incentives (e.g. treatment).

```r
model4_robust
```


```
##
## t test of coefficients:
##
##                   Estimate Std. Error t value  Pr(>|t|)
## (Intercept)       0.435216   0.055672  7.8175 3.122e-13 ***
## treatment_fruit -0.033972   0.080734 -0.4208   0.67437
## treatment_candy  0.028963   0.086491  0.3349   0.73808
## gender          -0.133656   0.071935 -1.8580   0.06465 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```


In the model including gender as a regressors, we can see that gender is significant at 10% and negative. As the gender dummy equals 1 for girlst, we can conclude that there is weakly significant effect and the girls tend to lie less than boys.

```r
model5 <- lm(correct ~ treatment_fruit + treatment_candy + gender*treatment_fruit + gender*treatment_can
model5_robust <- coeftest(model5, vcov = vcovHC, type = "HC1")
```

```
model5_robust
```

```
##
## t test of coefficients:
##
##                          Estimate Std. Error t value  Pr(>|t|)
## (Intercept)             0.4477612  0.0616733  7.2602 8.859e-12 ***
## treatment_fruit        -0.0331270  0.0995220 -0.3329   0.73959
## treatment_candy        -0.0284064  0.1090814 -0.2604   0.79482
## gender                 -0.1785304  0.1077143 -1.6574   0.09903 .
## treatment_fruit:gender -0.0008096  0.1691496 -0.0048   0.99619
## treatment_candy:gender  0.1591756  0.1790667  0.8889   0.37514
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(iv) Estimate your preferred model specification (including pupil characteristics if this improves your model) and write down your main conclusions. Prior to conducting the experiment, the researcher performed some power calculations.

treatments and gender, no interaction effect children just always lie

(v) Given the sample size and the estimates you have obtained above, what would be the minimum detectable effect size of this experiment?

```
# Fruit
k=4
q=0.8
alpha=0.05

n <- length(data$treatment)
m <- length(data$treatment[!data$treatment  == 0])

t_alpha <- qt(p=alpha/2, df=n-k, lower.tail=FALSE)
t_q <- qt(p=q, df=n-k, lower.tail=FALSE)
p = m/n

#Residual Standard error (Like Standard Deviation)
k=length(model4$coefficients)-1 #Subtract one to ignore intercept
SSE=sum(model4$residuals**2)
t=length(model4$residuals)
sigma = sqrt(SSE/(t-(1+k))) #Residual Standard Error

MDE = (t_alpha - t_q)*sqrt(1/(p*(1-p)))*sqrt(sigma^2/n)

MDE
```

```
## [1] 0.1936133
```

(vi) Initially, the researcher expected that no pupil would lie if there is nothing at stake (control group) and that a quarter of the student lie for a candy bar. How large should the sample size of the experiment have been in that case?

```
MDE_target = 0.25*(1 - (1/6))
samplesize = ((t_alpha - t_q)/MDE_target)^2*((sigma^2)/(p*(1-p)))
print(samplesize)
```

```
## [1] 175.3271
```

Note: same df used as above to not have to iteratively solve it (only marginal difference)

(vii) Students assigned to the control group provide counterfactuals to both the fruit and the candy bar treatment. Show that by assigning more students to the control group than to each treatment group the same MDE can be achieved with fewer students in the experiment.

```
p2 <- 0.5
new_size = ((t_alpha - t_q)/MDE)^2 * (sigma^2)/(p2*(1-p2))
new_size
```

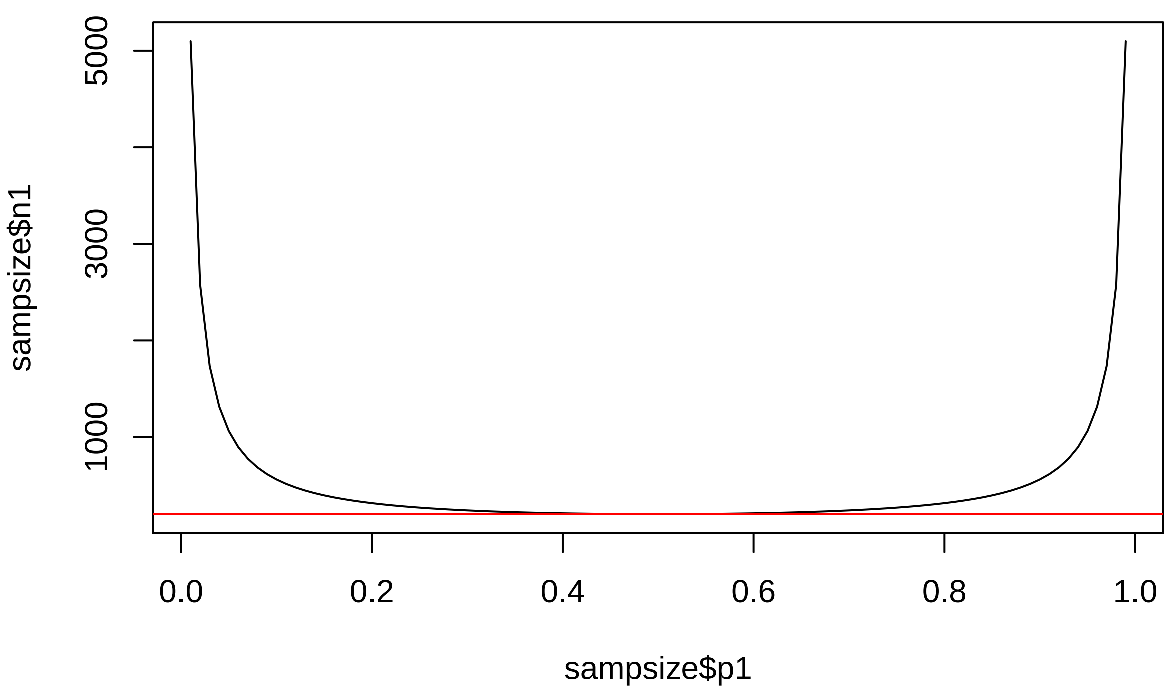
```
## [1] 201.8916
```

```
p1 <- seq.int(from = 0, to = 1, by=0.01)
n1 <- list()

for (i in 1:length(p1)){
  n1[i] = ((t_alpha - t_q)/MDE)^2 * (sigma^2)/(p1[i]*(1-p1[i]))
}

p1 <- as.list(p1)
sampsize = data.frame(unlist(p1),unlist(n1))
names(sampsize) <- c("p1", "n1")
sampsize[,2] <- as.numeric(sampsize[,2])
sampsize <- sampsize[c(-1,-101), ]

plot(sampsize$p1,sampsize$n1, type="l")
abline(h=length(data$treatment), col="red")
```



??? Assignment wants to increase treatment, but function is minimal for p=0.5, which also makes no sense, since then no one is in the control so there must be a fundamental mistake in how we calculate the stuff, uff