Assignment 3

David Gyaraki, Thao Le

Contents

1 Question 1 **2**
1.1 (i) . . . 2
1.2 (ii) . . . 2
1.3 (iii) . . . 3
1.4 (iv) . . . 4

2 Question 2 **4**
2.1 (i) . . . 4
2.2 (ii) . . . 6
2.3 (iii) . . . 8
2.4 (iv) . . . 8
2.5 (v) . . . 11
2.6 (vi) . . . 12
2.7 (vii) . . . 14

```r
# load packages
if(!require(pacman)){install.packages("pacman")}

p_load(devtools,tidyverse,dplyr,ggplot2,latex2exp,
       sampleSelection, quantreg, plm, nlme, knitr,car)
```

1 Question 1

color	number of individual		average outcome	
	treated	control	treated	control
purple	100	100	9	7
blue	75	25	13	8
green	25	75	10	9

1.1 (i)

The treatment effect in theory is the difference between the outcomes if the individual is treated versus if the individual is not treated. Suppose that for individual i, the treatment effect is defined as:

$$TE_i = \Delta_i = y_{i,d=1} - y_{i,d=0}, \tag{1}$$

where the y marks the outcome and d is the dummy whether individual i was treated ($d = 1$) or not ($d = 0$). However, this is rarely possible to be observed for each individual, as treatment is generally considered mutually exclusive, so an individual is either treated or not. Therefore, in this example we cannot calculate TE measures for individuals within color groups, but we can pretend as if one color corresponds to one observation, and calculate the three treatment effects across the color groups.

Then, the treatment effect for "observation" purple is $9 - 7 = 2$, the treatment effect for "observation" blue is $13 - 8 = 5$ and the treatment effect for "observation" green is $10 - 9 = 1$.

1.2 (ii)

Using the assumption of Section 1.1, we can infer here that we are looking for the average treatment effect of the full population including all colors. The average treatment effect is generally given by:

$$ATE = E[\Delta] = E[y_{d=1} - y_{d=0}], \tag{2}$$

where we take the average across the observation differences. Our data in the exercise does not contain information again on the individual outcomes, however we can use the average outcomes of treatment and control in each color group to calculate the total differences between $y_{d=1}$ and $y_{d=0}$ on the full population. Therefore, we can calculate the ATE as:

```
E_diff_purple = 9 - 7
E_diff_blue = 13 - 8
E_diff_green = 10 - 9

ATE = (E_diff_purple * 200 + E_diff_blue * 100 + E_diff_green * 100) /
 ↪  400
ATE
```

```
[1] 2.5
```

1.3 (iii)

Compared to the solution in Section 1.2, now we consider the average treatment effect among treated (ATET). For this case, we need to modify the previous definition to:

$$\begin{aligned} ATET &= E[\Delta|d=1] = E[y_{d=1} - y_{d=0}|d=1] = E[y_{d=1}] - E[y_{d=0}] \\ &= \frac{1}{N_T}\Sigma_{i=1}^{N_T} y_{i,d=1} - \frac{1}{N_{NT}}\Sigma_{i=1}^{N_{NT}} y_{i,d=0}, \end{aligned} \tag{3}$$

where N_T is the number of individuals in the treatment group while N_{NT} is the number of individuals in the control group. Therefore, the ATET for the whole population can be calculated as:

```
E_treatment_atet <- (9*100 + 13*75 + 10*25) / (100+75+25)
E_control_atet <- (7*100 + 8*25 + 9*75) / (100+75+25)

ATET = E_treatment_atet - E_control_atet
ATET
```

```
[1] 2.75
```

1.4 (iv)

The ATE measure generally describes the expected gain in y achieved by treating a random member i from the population, i.e. how much one benefits from being selected for the treatment compared to people who were not. On the other hand, ATET describes the average gain achieved by the treatment for the treated group, i.e. the comparison is not to the population but peer-to-peer, what is the expected benefit for those who are selected. The ATET is more helpful if we are not mainly interested in the potentially positive effect of the treatment for those who are treated versus those who are not, but rather the magnitude of these positive effects.

Suppose that we have an experiment where the government announces a new plan to introduce an additional level of health insurance, where the own risk cost would be cut in half, in order to investigate the effects of these on household savings. Initially, the government selects a few thousand people to have reduced own cost since it would be costly to select the whole population for treatment group. Arguably, cutting the own risk cost in half without changing the insurance monthly premiums would most likely have a positive effect on the wealth for those who are involved in the initial study, so the ATE estimate would likely tell us that program participants will have larger household savings due to the treatment. But if the government is rather interested in measuring the average savings surplus this would create for households, we would be more interested in the ATET measure. If the government wants this policy to be introduced for everyone later on, the main interest would be the estimated average savings surplus the treatment would create (for all households).

2 Question 2

```
dfData = read.csv("bonus.csv")
attach(dfData)

dfData <- na.omit(dfData)
```

2.1 (i)

```
df_noR <- dfData[dfData$bonus0 == 1,]
df_lowR <- dfData[dfData$bonus500 == 1,]
df_highR <- dfData[dfData$bonus1500 == 1,]

sum_noR <- sapply(df_noR, mean, na.rm=TRUE)
```

```r
sum_lowR <- sapply(df_lowR, mean, na.rm=TRUE)
sum_highR <- sapply(df_highR, mean, na.rm=TRUE)

summary_table <- cbind(sum_noR, sum_lowR, sum_highR)

kable(summary_table, caption="The means of the predictor and dependent
 ↪  variables across the three groups", col.names = c("no reward", "low
 ↪  reward", "high reward"), digits = 3)
```

Table 1: The means of the predictor and dependent variables across the three groups

	no reward	low reward	high reward
p0	0.558	0.525	0.576
job	0.760	0.840	0.811
stp2001	34.129	31.998	33.311
stp2004	89.183	82.268	86.311
dropout	0.360	0.346	0.324
myeduc	12.253	12.111	12.662
fyeduc	13.467	13.420	13.581
bonus0	1.000	0.000	0.000
bonus500	0.000	1.000	0.000
bonus1500	0.000	0.000	1.000
effort	19.549	18.273	18.483
pass	0.200	0.198	0.243
math	5.440	5.395	5.405

As we can see from the summary table above, the no reward group saw a 19.5% pass rate, the low reward group saw a 20.2% pass rate while the high reward group saw a 24.1% pass rate. At the same time, we can also look at the other numeric variables to investigate their means and check how balanced the other predictors are. Naturally since we grouped the students based on the incentives, the three *bonus* variables will not be balanced. However, the other ones are quite similar among the three groups implying that the background characteristics are relatively balanced. In particular, parental educational background (*myeduc* and *fyeduc*) and high-school math scores (*math*) are quite similar to one another among groups.

2.2 (ii)

```
prob_lm1 <- lm(pass ~ bonus500 + bonus1500, data = dfData)
prob_lm2 <- lm(pass ~ bonus500 + bonus1500 + fyeduc + p0 + math, data =
 ↪  dfData)
prob_lm3 <- lm(pass ~ bonus500 + bonus1500 + fyeduc + p0 + math +job
 ↪  +effort, data = dfData)
```

```
stargazer::stargazer(prob_lm1,prob_lm2,prob_lm3, title="Estimating the
 ↪  effects of treatment on first year pass rate",  align=TRUE, label =
 ↪  "tab_pass", table.placement="H", out = "tab_pass.tex")
```

Table 2: Estimating the effects of treatment on first year pass rate

	Dependent variable:		
	pass		
	(1)	(2)	(3)
bonus500	-0.002	0.011	0.023
	(0.066)	(0.059)	(0.058)
bonus1500	0.043	0.043	0.056
	(0.067)	(0.060)	(0.059)
fyeduc		-0.001	0.0003
		(0.007)	(0.007)
p0		0.238^{**}	0.170^{*}
		(0.098)	(0.098)
math		0.125^{***}	0.124^{***}
		(0.019)	(0.018)
job			-0.062
			(0.060)
effort			0.008^{***}
			(0.002)
Constant	0.200^{***}	-0.602^{***}	-0.678^{***}
	(0.048)	(0.131)	(0.144)
Observations	230	230	230
R^2	0.003	0.225	0.264
Adjusted R^2	-0.006	0.208	0.240
Residual Std. Error	0.412 (df = 227)	0.365 (df = 224)	0.358 (df = 222)
F Statistic	0.294 (df = 2; 227)	13.043^{***} (df = 5; 224)	11.359^{***} (df = 7; 222)
Note:			*p$<$0.1; **p$<$0.05; ***p$<$0.01

In order to estimate a model with the 2 treatment and 1 control groups, we need to be careful about the model specification. We cannot simply add all three dummy variables as predictors in the model, since the three groups are mutually exclusive and exhaustive so adding all three dummies in the model would cause perfect multicollinearity. Hence we can only use 2 of the 3

variables and the combination of both variables being 0 gives the "third variable". Therefore, we only estimate a model with the two treatment group dummies.

Looking at the summary statistics for the first model in Table 2 in column (1), we can see that the effects of the treatments are not statistically significant. Thus, we cannot say whether the treatment effect has an influence on the students' study achievements or not. This also implies that we cannot say much about the effect of being assigned in one of the treatment or control groups, as the lack of statistical significance hinders us from saying something about the sign of the coefficients, even though the coefficient of *bonus1500* is positive (implying that the treatment might increase the chance of passing all courses in first year).

In the second model, we add 3 additional control variables, namely, father's education *fyeduc*, subjective self-assessment of the pass probability *p0*, and high school math score *math*. The results are presented in Table 2 in column (2). However, the significance of the treatment variables *bonus500* and *bonus1500* do not change. Thus, this model also implies like the one above, that the financial incentive does not influence a student's study performance in the first year significantly.

2.3 (iii)

In the third model, we include the variable indicating whether a student has a job (*job*) and the average number of study hours (*effort*).

To comment on this approach, we do not think these two control variables are good addition to the model as it might be correlated with the treatment variables and with each other. For example, the amount of study effort one puts in might correlate with the type of reward they are assign (e.g., a higher reward means more study efforts). On the other hand, if the reward of studying is low, a student might be more incentivised to take up a part-time job, and the fact that a student has a job might lower the average number of hours they put into the study (which is measures in the variable *effort*).

Nevertheless, the model still results in insignificant coefficients for the two treatment dummy variables as displayed in Table 2 in column (3), implying that the two added variables compared to Section 2.2 might indeed cause multicollinearity issues as the insignificance might also be attributed to imploded standard errors.

2.4 (iv)

For this question, we use a linear probability model similar to Section 2.3, where we aim to explain the outcome variable with the variables *bonus500*, *bonus1500*, *math*, *fyeduc*, *p0*, *effort* and *job* (for explanation of these variables, see Section 2.2 and Section 2.3). Hence, we build 3 linear probability models that include the treatment variables and all exogenous variables as the predictors.

```r
drop_lm <- lm(dropout ~ bonus500 + bonus1500 + math + fyeduc + p0 +
 ↪  effort + job, data=dfData)
credsYear1_lm <- lm(stp2001 ~ bonus500 + bonus1500 + math + fyeduc + p0
 ↪  + effort + job, data=dfData)
credsYear3_ln <- lm(stp2004 ~ bonus500 + bonus1500 + math + fyeduc + p0
 ↪  + effort + job, data=dfData)
```

```r
stargazer::stargazer(drop_lm,credsYear1_lm,credsYear3_ln,
 ↪  title="Estimating dropout and credit rates with the previous model",
 ↪  align=TRUE, table.placement="H", label="tab_dropcred", out =
 ↪  "tab_dropcred.tex")
```

Table 3: Estimating dropout and credit rates with the previous model

	Dependent variable:		
	dropout	stp2001	stp2004
	(1)	(2)	(3)
bonus500	-0.048	-0.144	-2.388
	(0.068)	(2.690)	(7.999)
bonus1500	-0.057	0.227	0.294
	(0.070)	(2.744)	(8.160)
math	-0.077^{***}	6.565^{***}	15.561^{***}
	(0.022)	(0.856)	(2.545)
fyeduc	0.009	-0.309	-1.231
	(0.009)	(0.337)	(1.002)
p0	-0.150	13.582^{***}	18.312
	(0.116)	(4.567)	(13.579)
effort	-0.018^{***}	0.928^{***}	2.920^{***}
	(0.003)	(0.111)	(0.331)
job	0.034	-0.904	5.628
	(0.071)	(2.791)	(8.298)
Constant	1.071^{***}	-22.455^{***}	-50.458^{**}
	(0.170)	(6.713)	(19.962)
Observations	230	230	230
R^2	0.232	0.445	0.388
Adjusted R^2	0.207	0.427	0.369
Residual Std. Error (df = 222)	0.424	16.695	49.642
F Statistic (df = 7; 222)	9.563^{***}	25.388^{***}	20.112^{***}
Note:			*p<0.1; **p<0.05; ***p<0.01

The summary shows a clear picture. For all three dependent variables, *dropout* as a measure of whether the student dropped out of the program in model (1) in Table 3, *stp2001* and *stp2004* measuring the number of credits collected in the first and in the three years respectively (estimated in model (2) and (3) in Table 3), only *math* and *effort* being significant, with

p0 significant only for model (2). Thus good high school math score implies lower risk of dropping out and higher amount of credits collected for both first and all years. Furthermore, the amount of effort (number of hours on average) works in a similar fashion, i.e. the more the student studies, the less likely they will be to drop out and the more credits they will achieve. Interestingly, the self-assessed likelihood of success before the program also seems to impact the first year credits positively, which might mean that there is an underlying variable such as motivation connected to this. However, we can also concur that the effect of the financial incentives is not significant, as both *bonus500* and *bonus1500* are insignificant for all three models.

2.5 (v)

The minimum detectable effect of this experiment is the measure using the formula:

$$MDE = (t_{1-\alpha/2} - t_{1-q})\sqrt{\frac{1}{p(1-p)}}\sqrt{\frac{\sigma^2}{n}} \tag{4}$$

We need to calculate the MDE for both low-reward versus control group and high-reward versus control group. Therefore, our calculations follow as:

```
MinDE <- function(dfData, input_model, sTreatment, dAlpha, dPower){
  num_coef <- length(input_model$coefficients)

  # get the number of observations
  n_obs <- nrow(dfData)
  # Observations in treated group
  n_treatment <- sum(dfData[,sTreatment])
  # Proportion of treatment observations
  proportion <- n_treatment/n_obs

  # T_statistics of alpha and power levels
  t_stats_alpha <- qt(1-dAlpha/2, n_obs - num_coef)
  t_power <- qt(1-dPower, n_obs - num_coef)

  # get variance of residuals
  dVariance <- var(input_model$residuals)

  # get the MDE
  MDE <- (t_stats_alpha - t_power) * sqrt(1/(proportion*(1-proportion)))
↪  * sqrt(dVariance/n_obs)
```

```
  return(MDE)
}

# Initialize desired alpha and power
dAlpha = 0.05
dPower = 0.8

# Prepare dataframe
df_low_treatment = dfData[dfData$bonus1500 != 1,]
df_high_treatment = dfData[dfData$bonus500 != 1,]

#Estimate linear model with low reward treatment and control group

Low_treatment = lm(pass ~ bonus500 + math + fyeduc + p0 + effort + job,
 ↪  data=df_low_treatment)

#Get MDE of low-reward versus control group
MDE_low = MinDE(df_low_treatment,Low_treatment,"bonus500",dAlpha,
 ↪  dPower)

#Estimate linear model with high reward treatment and control group
High_treatment = lm(pass ~ bonus1500 + math + fyeduc + p0 + effort +
 ↪  job, data=df_high_treatment)

#Get MDE of high-reward versus control group
MDE_high = MinDE(df_high_treatment, High_treatment,"bonus1500",dAlpha,
 ↪  dPower)

cat("The MDE of the low reward treatment group is: ", MDE_low, ", \n",
 ↪  "and the MDE of the high reward treatment group is: ", MDE_high)
```

```
The MDE of the low reward treatment group is:  0.1577769 ,
 and the MDE of the high reward treatment group is:  0.1679406
```

2.6 (vi)

An increases in the pass rate of 10% points correspond with the Minimum Detectable Effect size of 10%. Given this, we can calculate the minimum number of observations needed using the rewritten version of Equation (4):

$$n = \left(\frac{t_{1-\alpha/2} - t_{1-q}}{MDE} \right)^2 \frac{\sigma^2}{p(1-p)} \tag{5}$$

However, for this we need to make a few assumptions. Since we are looking for the minimum sample size, we do not know n and the degrees of freedom to calculate the t-stats in the numerator of Equation (5). Therefore, we need to assume that n is large enough to approach normal distribution (asymptotics) and calculate the statistics using a standard normal distribution.

```r
get_n <- function(dfData, sTreatment, dAlpha, dPower, MinDE=0.1,
 ↪  dSigma_sq){
  # T_statistics of alpha and power levels
  t_stats_alpha <- qnorm(1-dAlpha/2, 0,dSigma_sq)
  t_power <- qnorm(1-dPower, 0, dSigma_sq)

  # get the number of observations
  n_obs <- nrow(dfData)
  # Observations in treated group
  n_treatment <- sum(dfData[,sTreatment])
  # Proportion of treatment observations
  proportion <- n_treatment/n_obs

  # get the desired sample size
  sample_size = ((t_stats_alpha-t_power)/MinDE)^2
 ↪  *(dSigma_sq/(proportion*(1-proportion)))
  return(round(sample_size))
}

# Initialize, set sigma and alpha fixed
dSigma_sq = 1
dAlpha= 0.05
dPower = 0.8

# required sample sizes needed in each treatment group
sample_low_required = get_n(df_low_treatment, "bonus500", dAlpha =
 ↪  dAlpha, dPower = dPower, MinDE=0.1, dSigma_sq = dSigma_sq)

sample_high_required = get_n(df_high_treatment, "bonus1500", dAlpha =
 ↪  dAlpha, dPower = dPower, MinDE=0.1, dSigma_sq = dSigma_sq)
```

```r
cat("The minimum sample size to detect a 10% point increase", "\n", "in
 ↪  the low reward treatment group is: ", sample_low_required, ", \n",
 ↪  "in the high reward treatment group is: ", sample_high_required, ",
 ↪  \n", "and the number of observations in the dataset is:", "\n", "in
 ↪  the control and low treatment: ", nrow(df_low_treatment), ", \n",
 ↪  "and in the control and high treatment group: ",
 ↪  nrow(df_high_treatment))
```

```
The minimum sample size to detect a 10% point increase
 in the low reward treatment group is:  3144 ,
 in the high reward treatment group is:  3140 ,
 and the number of observations in the dataset is:
 in the control and low treatment:  156 ,
 and in the control and high treatment group:  149
```

From these results, we can establish that the sample size of the data is much lower than the required minimum sample size to be able to detect a 10% increase in the pass rates. Hence perhaps it would be interesting to see what would happen to the treatment coefficients if one would considerably increase the sample size while keeping the treatment and control groups balanced.

2.7 (vii)

Looking at the formula for MDE in Equation (4), fewer students in the experiment mean higher MDE (since n is in denominator). If we want to achieve the same MDE with fewer students, we need to choose the proportion of control versus treatment group such that the term $p(1-p)$ is minimized (since this term is also in the denominator). Taking the derivative of that term with respect to p, we find the optimal value of $p = 0.5$ where p is the proportion of treated students. To get $p = 0.5$, we indeed need to increase number of students in the control group.

In this case, we consider a different case from the one in Section 2.6: since control provides counterfactual to both low and high treatment, we will consider the proportion between control and the sum of the two treatment groups. In our data, this means that the size of the control group needs to be equal to the size of the treatment groups such that we can achieve the same MDE with lower minimum sample size. First let us look at the current minimum sample size such that the treatment groups are considered as one. Then we can calculate with the same number of observations what if the control group would make up half of the observations.

```r
get_n_eq <- function(dfData, dT, dAlpha, dPower, MinDE=0.1, dSigma_sq){
  # T_statistics of alpha and power levels
  t_stats_alpha <- qnorm(1-dAlpha/2, 0,dSigma_sq)
  t_power <- qnorm(1-dPower, 0, dSigma_sq)

  # get the number of observations
  n_obs <- nrow(dfData)
  # Observations in treated group
  n_treatment <- dT
  # Proportion of treatment observations
  proportion <- n_treatment/n_obs

  # get the desired sample size
  sample_size = ((t_stats_alpha-t_power)/MinDE)^2
  *(dSigma_sq/(proportion*(1-proportion)))
  return(round(sample_size))
}

# Initialize, set sigma and alpha fixed
dSigma_sq = 1
dAlpha= 0.05
dPower = 0.8
dT = nrow(dfData[dfData$bonus0 == 0,])

dTideal = nrow(dfData)/2

# required sample sizes needed in each treatment group
sample_required_eq = get_n_eq(dfData=dfData, dT=dT, dAlpha = dAlpha,
  dPower = dPower, MinDE=0.1, dSigma_sq = dSigma_sq)

sample_required_contr = get_n_eq(dfData=dfData, dT=dTideal, dAlpha =
  dAlpha, dPower = dPower, MinDE=0.1, dSigma_sq = dSigma_sq)

cat("The minimum sample size required for MDE=10% implied by dataset: ",
  sample_required_eq, ", \n", "and the minimum sample size with equal
  proportions:", sample_required_contr)
```

```
The minimum sample size required for MDE=10% implied by dataset:  3572 ,
 and the minimum sample size with equal proportions: 3140
```

This result implies that if we improved the proportions of the control and treatment groups

by increasing the control group size to be above both the low and high treatment, we could achieve a large reduction in the minimum sample size required for an MDE of 10%, namely from 3572 to 3140. If we consider the control to be counterfactual to both treatment groups then, we could reduce the required students in the experiment.