Econometrics II TA Session #3

Hiroki Kato

1 Empirical Application of Binary Model: Racial Discrimination in Court

Brief Background. Recently, in the U.S., anti-racism activities called "Black Lives Matter" are getting hot. These activities stems from the death of George Floyd, who was killed by a white police officer on May 25, 2020. The empirical application of binary model investigates whether the judgement of death penalty is based on race of defendant and race of victim.

Data. The package `catdata` contains many built-in dataets which include categorical variables. We use the built-in dataset `deathpenalty` which is about the death-penalty judgement of defendants in cases of multiple murders in Florida between 1976 and 1987.

```r
# If there is no package called 'catdata', run 'install.packages("catdata")'
# After that run following codes
library(catdata)
data(deathpenalty)
deathpenalty
```

```
##   DeathPenalty VictimRace DefendantRace Freq
## 1            0          0             0  139
## 2            1          0             0    4
## 3            0          1             0   37
## 4            1          1             0   11
## 5            0          0             1   16
## 6            1          0             1    0
## 7            0          1             1  414
## 8            1          1             1   53
```

This dataset contains three dummy variables

1. `DeathPenalty` is a dummy variable taking 1 if the judgement is death penalty.
2. `VictimRace` is a dummy variable taking 1 if the race of the victim is white.
3. `DefendantRace` is a dummy variable taking 1 if the race of the defendant is white.

This dataset aggregates observations with repect to `DeathPenalty`, `VictimRace` and `DefendantRace`. The variable `Freq` represents the number of observations. Since it is inconvinient for us to use the original data for estimation, we disaggregate this dataset. For example, we make 37 rows whose elements are `DeathPenalty = 0`, `VictimRace = 1`, and `DefendantRace = 0` because there are 37 obaservations, i.e., `Freq = 37`.

```
dt <- deathpenalty
dt <- dt[rep(seq_len(nrow(dt)), dt[,"Freq"]), -4]
```

Model. In a binary model, a dependent (outcome) variable y_i takes only two values, i.e., $y_i \in \{0, 1\}$. A binary variable is sometimes called a *dummy* variable. In this application, the outcome variable is `DeathPenalty` taking 1 if the judgement is death penalty. We make three explanatory variables.

1. `WB` is a dummy variable taking 1 if the race of the victim and the defendant is white and black, respectively.
2. `BW` is a dummy variable taking 1 if the race of the victim and the defendant is black and white, respectively.
3. `WW` is a dummy variable taking 1 if the race of both the victim and the defendant is black.

```
dt$WB <- ifelse(dt$VictimRace == 1 & dt$DefendantRace == 0, 1, 0)
dt$BW <- ifelse(dt$VictimRace == 0 & dt$DefendantRace == 1, 1, 0)
dt$WW <- ifelse(dt$VictimRace == 1 & dt$DefendantRace == 1, 1, 0)
```

The regression function is

$$
\begin{aligned}
&\mathbb{E}[DeathPenalty|WB, BW, WW] \\
=&\mathbb{P}[DeathPenalty = 1|WB, BW, WW] = G(\beta_0 + \beta_1 WB + \beta_2 BW + \beta 3WW).
\end{aligned}
$$

The function $G(\cdot)$ is arbitrary function. In practice, we often use following three specifications:

- Linear probability model (LPM): $G(\mathbf{x}_i\beta) = \mathbf{x}_i\beta$.
- Probit model: $G(\mathbf{x}_i\beta) = \Phi(\mathbf{x}_i\beta)$ where $\Phi(\cdot)$ is the standard Gaussian cumulative function.
- Logit model: $G(\mathbf{x}_i\beta) = 1/(1 + \exp(-\mathbf{x}_i\beta))$.

1.1 Linear Probability Model

The linear probability model is

$$
\mathbb{P}[DeathPenalty = 1|WB, BW, WW] = \beta_0 + \beta_1 WB + \beta_2 BW + \beta 3WW
$$

This model can be estimated using the OLS method. In `R`, we can use the OLS method, running `lm()` function.

```
model <- DeathPenalty ~ WB + BW + WW
LPM <- lm(model, data = dt)
```

However, `lm()` function does not deal with heteroskedasticity problem. To resolve it, we need to claculate heteroskedasticity-robust standard errors using the White method.

$$\hat{V}(\hat{\beta}) = \left(\frac{1}{n}\sum_i \mathbf{x}_i'\mathbf{x}_i\right)^{-1} \left(\frac{1}{n}\sum_i \hat{u}_i^2\mathbf{x}_i'\mathbf{x}_i\right) \left(\frac{1}{n}\sum_i \mathbf{x}_i'\mathbf{x}_i\right)^{-1}$$

```r
# heteroskedasticity-robust standard errors
dt$"(Intercept)" <- 1
X <- as.matrix(dt[,c("(Intercept)", "WB", "BW", "WW")])
u <- diag(LPM$residuals^2)

XX <- t(X) %*% X
avgXX <- XX * nrow(X)^{-1}
inv_avgXX <- solve(avgXX)

uXX <- t(X) %*% u %*% X
avguXX <- uXX * nrow(X)^{-1}

vcov_b <- (inv_avgXX %*% avguXX %*% inv_avgXX) * nrow(X)^{-1}
rse_b <- sqrt(diag(vcov_b))

# homoskedasticity-based standard errors
se_b <- sqrt(diag(vcov(LPM)))

print("The Variance of OLS"); vcov(LPM)
```

```
## [1] "The Variance of OLS"

##               (Intercept)            WB            BW            WW
## (Intercept)  0.0006194791 -0.0006194791 -0.0006194791 -0.0006194791
## WB          -0.0006194791  0.0024650104  0.0006194791  0.0006194791
## BW          -0.0006194791  0.0006194791  0.0061560732  0.0006194791
## WW          -0.0006194791  0.0006194791  0.0006194791  0.0008091697
```

```r
print("The Robust variance of OLS"); vcov_b
```

```
## [1] "The Robust variance of OLS"

##              (Intercept)           WB           BW            WW
## (Intercept)  0.000190137 -0.000190137 -0.000190137 -0.0001901370
## WB          -0.000190137  0.003870331  0.000190137  0.0001901370
## BW          -0.000190137  0.000190137  0.000190137  0.0001901370
## WW          -0.000190137  0.000190137  0.000190137  0.0004055766
```

```r
print("The Robust se using White method"); rse_b
```

```
## [1] "The Robust se using White method"
```

```
## (Intercept)          WB          BW          WW
##  0.01378902  0.06221198  0.01378902  0.02013893
```

```
print("The Robust t-value using White method"); coef(LPM)/se_b
```

```
## [1] "The Robust t-value using White method"
```

```
## (Intercept)          WB          BW          WW
##   1.1238559   4.0523507  -0.3565103   3.0063494
```

Using the package `lmtest` and `sandwich` is the most easiest way to calculate heteroskedasticity-robust standard errors and t-statistics.

```
library(lmtest) #use function `coeftest`
library(sandwich) #use function `vcovHC`
coeftest(LPM, vcov = vcovHC(LPM, type = "HC0"))[, "Std. Error"]
```

```
## (Intercept)          WB          BW          WW
##  0.01378902  0.06221198  0.01378902  0.02013893
```

```
coeftest(LPM, vcov = vcovHC(LPM, type = "HC0"))[, "t value"]
```

```
## (Intercept)          WB          BW          WW
##    2.028573    3.234017   -2.028573    4.246418
```

Finally, we obtain follwing results of linear probability model. We will discuss interpretation of results and goodness-of-fit of LPM later.

```
# t-stats
t_b <- coef(LPM)/se_b
rt_b <- coef(LPM)/rse_b
# p-value Pr( > |t|)
p_b <- pt(abs(t_b), df = nrow(X)-ncol(X), lower = FALSE)*2
rp_b <- pt(abs(rt_b), df = nrow(X)-ncol(X), lower = FALSE)*2

library(stargazer)
stargazer(
  LPM, LPM,
  se = list(se_b, rse_b), t = list(t_b, rt_b), p = list(p_b, rp_b),
  t.auto = FALSE, p.auto = FALSE,
  report = "vcstp", keep.stat = c("n"),
  add.lines = list(
    c("Standard errors", "Homoskedasticity-based", "Heteroskedasticity-robust")),
  title = "Results of Linear Probability Model",
  type = "latex", header = FALSE, font.size = "small"
)
```

Table 1: Results of Linear Probability Model

	Dependent variable:	
	DeathPenalty	
	(1)	(2)
WB	0.201	0.201
	(0.050)	(0.062)
	t = 4.052	t = 3.234
	p = 0.0001	p = 0.002
BW	−0.028	−0.028
	(0.078)	(0.014)
	t = −0.357	t = −2.029
	p = 0.722	p = 0.043
WW	0.086	0.086
	(0.028)	(0.020)
	t = 3.006	t = 4.246
	p = 0.003	p = 0.00003
Constant	0.028	0.028
	(0.025)	(0.014)
	t = 1.124	t = 2.029
	p = 0.262	p = 0.043
Standard errors	Homoskedasticity-based	Heteroskedasticity-robust
Observations	674	674
Note:		*p<0.1; **p<0.05; ***p<0.01