HW4: Panel Data

Peng Peng

4/3/2019

Exercise 1 Data

```
# Randomly select 5 observations from the dataset-------------------------

# Calculate number of observations for each person
n = data %>%
    group_by(PERSONID) %>%
    count() %>%
    select(n)

# Convert a dataframe to vector
m = as.matrix(n)

# Create a nested data frame and add number of observations for each indivdiual
d_nested = data %>%
           group_by(PERSONID) %>%
           nest() %>%
           mutate(n = m)

# Randomly select 5 individuals from the list frame and unnest
d_sample = sample_n(d_nested, 5, replace = FALSE) %>%
           unnest()

# Plot the log wage across time periods for individiuals to show panel dimension

gg_panel = ggplot(d_sample,
                  aes(x = TIMETRND,
                      y = LOGWAGE,
                      color = as.factor(PERSONID))) +
           geom_jitter() +
           xlab("Time Trend") + ylab("Log Wage") +
           theme(legend.position = "bottom")
```

Exercise 2 Random Effects Model

```
library(lme4)

model_re = lmer(LOGWAGE ~ EDUC + POTEXPER + (1|PERSONID),
                data = data,
                REML = TRUE)

model_re
```

```
## Linear mixed model fit by REML ['lmerMod']
```

```
## Formula: LOGWAGE ~ EDUC + POTEXPER + (1 | PERSONID)
##    Data: data
## REML criterion at convergence: 16700.72
## Random effects:
##  Groups   Name        Std.Dev.
##  PERSONID (Intercept) 0.3647
##  Residual             0.3360
## Number of obs: 17919, groups:  PERSONID, 2178
## Fixed Effects:
## (Intercept)         EDUC     POTEXPER
##     0.56679      0.10771      0.03876
```

Exercise 3 Fixed Effects Model

```
# Between estimator -----------------------------------------------------

# Compute averages of Xs and Y
d_bt = data %>% group_by(PERSONID) %>%
       mutate(log_wage_m = mean(LOGWAGE)) %>%
       mutate(edu_m = mean(EDUC)) %>%
       mutate(exp_m = mean(POTEXPER))

# Regress mean(y) agains mean(Xs)
model_between = lm(log_wage_m ~ edu_m + exp_m, data = d_bt)


# Within estimator--------------------------------------------------------

# Compute differences in Xs and Y within cross-sectional data
d_wt  = d_bt %>% ungroup() %>%
         mutate(log_wage_diff = log_wage_m - LOGWAGE) %>%
         mutate(edu_diff = edu_m - EDUC) %>%
         mutate(exp_diff = exp_m - POTEXPER)

# Regress time-demeaned y against time-demaned Xs
model_within = lm(log_wage_diff ~ 0 + edu_diff + exp_diff, data = d_wt)

# First difference estimator------------------------------------------

# Compute differences in time periods and only select first differences
d_fd = data %>%
      group_by(PERSONID) %>%
      mutate(first_diff = TIMETRND - lag(TIMETRND)) %>%
      filter(first_diff == 1)

d_fd = d_fd %>%
       mutate(log_wage_fd = LOGWAGE - lag(LOGWAGE)) %>%
       mutate (edu_fd = EDUC - lag(EDUC)) %>%
       mutate(exp_fd = POTEXPER - lag(POTEXPER))

# Regress ydiff against X first diff
model_firstdiff = lm(log_wage_fd ~ 0 + edu_fd + exp_fd, data = d_fd)
```

```r
# Compare coefficients across models
estimate = rbind(model_between$coefficients[2:3], model_within$coefficients, model_firstdiff$coefficien
names(estimate) = c("intercept", "beta_education", "beta_potexper")
rownames(estimate) = c("between", "within", "first difference")
estimate
```

```
##                        edu_m      exp_m
## between          0.08767487 0.03027864
## within           0.12366202 0.03856107
## first difference 0.09784602 0.03421722
## attr(,"names")
## [1] "intercept"      "beta_education" "beta_potexper"  NA
## [5] NA               NA
```

Exercise 4 Understanding Fixed Effects

```r
# Likelihood function of fixed effects-------------------------------------

# Select 100 individuals
d_sample_100 = sample_n(d_nested, 100, replace = FALSE) %>%
               unnest()


library(fastDummies)
indicator = dummy_cols(d_sample_100, select_columns = "PERSONID")
indicator = indicator[, grepl("PERSONID_", colnames(indicator))]
indicator = indicator[, -1]


# Write the likelihood function
x = as.matrix(d_sample_100[, c("EDUC", "POTEXPER")])
y = as.matrix(d_sample_100$LOGWAGE)


LL = function(c){
  X = x
  Y = y
  beta = c[2:length(c)]
  sigma2 = c[1]
  n = nrow(x)
  ll = - (n/2) * log(2 * pi) - (n/2) * log(sigma2) - (1/(2 * sigma2)) * sum((y - x %*% beta)^2)
  return(ll)
}

# Optimize the likelihood function
x = cbind(as.matrix(x), as.matrix(indicator))
b = rnorm(102)
set.seed(1)
fit = optim(par = b, LL)
par = matrix(fit$par)
```

```r
# Regress individual FE against invariant variables---------------------
par = par[4: length(par)]
par = c(0,par)

# Calculate individual FE
d_sample_fe =
d_sample_100 %>% group_by(PERSONID) %>% filter(row_number()==1) %>% select(1:11) %>% select(-n) %>% as.m
d_sample_fe = data.frame(cbind(d_sample_fe, par))


model_fe_100 = lm(par ~ ABILITY + MOTHERED + FATHERED + BRKNHOME + SIBLINGS, data = d_sample_fe)

model_fe_100$coefficients
```

```
## (Intercept)     ABILITY    MOTHERED    FATHERED    BRKNHOME    SIBLINGS
##  0.28663579  0.05014743  0.07694577 -0.09306053 -0.39408640 -0.02933265
```

```r
# Standard errors ---------------------------------------------------
# The standard errors are incorrect because in OLS,
# we are assuming that error terms are normally
# distributed and independent of each other.
# However, by introducing fixed effects,
# the composite error terms are not
# independent of each other and thus not normally distributed.

# Correct standard errors: Huber-White sandwich formula

x = as.matrix(data[, c("EDUC", "POTEXPER")])
inv_x = solve(t(x) %*% x)
res = model_within$residuals
D = t(x) %*% diag(res)^2 %*% x
EHW = inv_x %*% D %*% inv_x
diag(sqrt(EHW))
```

```
##         EDUC     POTEXPER
## 0.0004080221 0.0005525334
```