ECON-613 HW #4

Peter Kim

2019-04-16

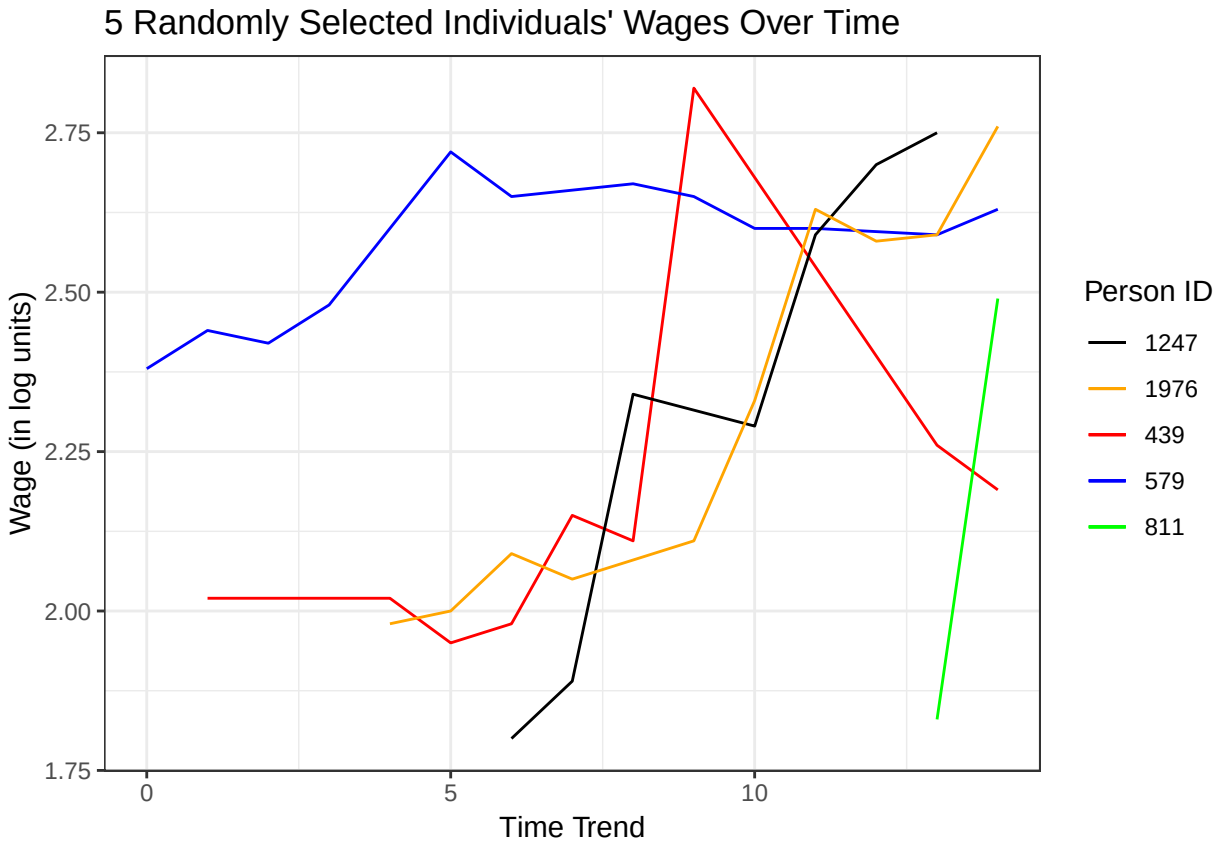
Exercise 1 Data

```
#Set Seed
set.seed(1)

#Read Data
data = read_csv("~/ECON-613/Assignment #4/HW #4/Koop-Tobias.csv")
```

```
## Parsed with column specification:
## cols(
##   PERSONID = col_double(),
##   EDUC = col_double(),
##   LOGWAGE = col_double(),
##   POTEXPER = col_double(),
##   TIMETRND = col_double(),
##   ABILITY = col_double(),
##   MOTHERED = col_double(),
##   FATHERED = col_double(),
##   BRKNHOME = col_double(),
##   SIBLINGS = col_double()
## )
```

```
#Identifying Unique Person ID
uniq_ID = unique(data$PERSONID) %>% length()

#Generating 5 Random Indices
ind = sample(x = 1:uniq_ID, size = 5, replace = FALSE)

#Representing the Panel Dimension of Wages for 5 Randomly Selected Individuals
rand5 = data %>%
  filter(PERSONID %in% ind) %>%
  select(PERSONID, TIMETRND, LOGWAGE)

rand5.1 = rand5 %>% filter(PERSONID == unique(rand5$PERSONID)[1])
rand5.2 = rand5 %>% filter(PERSONID == unique(rand5$PERSONID)[2])
rand5.3 = rand5 %>% filter(PERSONID == unique(rand5$PERSONID)[3])
rand5.4 = rand5 %>% filter(PERSONID == unique(rand5$PERSONID)[4])
rand5.5 = rand5 %>% filter(PERSONID == unique(rand5$PERSONID)[5])

ggplot() +
  geom_line(data = rand5.1,
            mapping = aes(x = TIMETRND, y = LOGWAGE, color = "439")) +
  geom_line(data = rand5.2,
            mapping =  aes(x = TIMETRND, y = LOGWAGE, color = "579")) +
  geom_line(data = rand5.3,
            mapping =  aes(x = TIMETRND, y = LOGWAGE, color = "811")) +
```

```
  geom_line(data = rand5.4,
            mapping =  aes(x = TIMETRND, y = LOGWAGE, color = "1247")) +
  geom_line(data = rand5.5,
            mapping =  aes(x = TIMETRND, y = LOGWAGE, color = "1976")) +
  scale_color_manual(
    name = "Person ID",
    values = c("439" = "red",
               "579" = "blue",
               "811" = "green",
               "1247" = "black",
               "1976" = "orange")
  ) +
  labs(
    x = "Time Trend",
    y = "Wage (in log units)",
    title = "5 Randomly Selected Individuals' Wages Over Time"
  ) +
  theme_bw()
```




```
#Alternative Way to Represent the Panel Dimension of Wages for 5 Randomly Selected Individuals
panel5 = data %>%
  filter(PERSONID %in% unique(rand5$PERSONID)) %>%
  group_by(PERSONID) %>%
  summarise(Size = n())

kable(panel5,
      caption = "Frequency of 5 Randomly Selected Individuals Showed Up in Data")
```

Table 1: Frequency of 5 Randomly Selected Individuals Showed Up in Data

PERSONID	Size
439	9
579	13
811	2
1247	7
1976	11

Comments:

I have represented the panel dimension of wages for 5 randomly selected individuals by a graph and a table. The graph is designed to present how the wage, in log units, is changing over time trend for each of the five inidividuals. The table is included to imply that the the panel data is unbalanced.

Exercise 2 Random Effects

```
#Implementing Linear Regresson
randlm = lme4::lmer(LOGWAGE ~ EDUC + POTEXPER + (1|PERSONID), data = data)

randlm_coef = randlm@beta %>% as.data.frame()
colnames(randlm_coef) = c("OLS Coefficients")
rownames(randlm_coef) = c("Intercept", "EDUC", "POTEXPER")

kable(randlm_coef, digits = 4,
      caption = "Table of OLS Coefficients for Random Effects Model")
```

Table 2: Table of OLS Coefficients for Random Effects Model

	OLS Coefficients
Intercept	0.5668
EDUC	0.1077
POTEXPER	0.0388

Comment

$\hat{\beta}_{intercept} = 0.5668$: When education and potential experience are 0, log wage will be 0.5668. This quantity, however, is meaningless since no individuals are legally permitted to have 0 units of education.

$\hat{\beta}_{EDUC} = 0.1077$: Ceteris paribus, one unit increase in education will increase log wage by 0.1077 on average.

$\hat{\beta}_{POTEXPER} = 0.0388$: Ceteris paribus, one unit increase in potential experience will increase log wage by 0.0388 on average.

Exercise 3 Fixed Effects Model

Between Estimator

We calculate between estimators here and save them as between_coef.

```
between_data = data %>%
  select(PERSONID, EDUC, LOGWAGE, POTEXPER, TIMETRND) %>%
  group_by(PERSONID) %>%
  summarise(mean_LOGWAGE = mean(LOGWAGE),
            mean_EDUC = mean(EDUC),
            mean_POTEXPER = mean(POTEXPER))

between_coef =
  lm(mean_LOGWAGE ~ mean_EDUC + mean_POTEXPER, data = between_data) %>%
  coefficients() %>%
  as.data.frame()
```

Within Estimator

We calculate within estimators here and save them as within_coef.

```
#Creating Initial Within Data
within_data = left_join(data, between_data, by = "PERSONID")

#Updating Within Data - Adding Columns of Y - Y_bar and X - X_bar
within_data2 = within_data %>%
  mutate(bet_resp = LOGWAGE - mean_LOGWAGE,
         bet_EDUC = EDUC - mean_EDUC,
         bet_POTEXPER = POTEXPER - mean_POTEXPER) %>%
  select(PERSONID, bet_resp, bet_EDUC, bet_POTEXPER)

#Calculating Within Coefficients
within_coef =
  lm(bet_resp ~ bet_EDUC + bet_POTEXPER - 1, data = within_data2) %>%
  coefficients() %>%
  as.data.frame()
```

First Time Difference Estimator

We calculate first time difference estimators here and save them as first_coef.

```
#Creating First Difference Data
first_data =
  data %>%
  group_by(PERSONID) %>%
  mutate(LOGWAGE_Diff = LOGWAGE - lag(LOGWAGE),
           EDUC_Diff = EDUC - lag(EDUC),
           POTEXPER_Diff = POTEXPER - lag(POTEXPER)) %>%
  select(PERSONID, LOGWAGE_Diff, EDUC_Diff, POTEXPER_Diff) %>%
  na.omit()

first_coef =
```

```
  lm(LOGWAGE_Diff ~ EDUC_Diff + POTEXPER_Diff - 1, data =  first_data) %>%
  coefficients() %>%
  as.data.frame()
```

Here, we create a table of $\hat{\beta}_{EDUC}$ and $\hat{\beta}_{POTEXPER}$

```
coef_data = data.frame(
  c(between_coef %>% unlist() %>% as.numeric()),
  c(NA, within_coef %>% unlist() %>% as.numeric()),
  c(NA, first_coef %>% unlist() %>% as.numeric)
)
colnames(coef_data) = c("Between", "Within", "First")
rownames(coef_data) = c("Intercept", "Education", "POTEXPER")

kable(coef_data, digits = 4,
      caption = "Coefficients Under Different Models")
```

Table 3: Coefficients Under Different Models

	Between	Within	First
Intercept	0.8456	NA	NA
Education	0.0931	0.1237	0.0479
POTEXPER	0.0260	0.0386	0.0329

Comparison of $\hat{\beta}_{education}$'s:

We observe that $\hat{\beta}_{between}$, $\hat{\beta}_{within}$, and $\hat{\beta}_{first}$ are all positive. Each model believes that a unit increase in education will increase wage (or log wage). The "within" model has the largest coefficient magnitude, while the "first-difference" model has the smallest coefficient magnitude.

Comparison of $\hat{\beta}_{POTEXPER}$'s:

Similar to education, we observe that $\hat{\beta}_{between}$, $\hat{\beta}_{within}$, and $\hat{\beta}_{first}$ are all positive. Each model believes that a unit increase in potential experience will increase wage (or log wage).. The "within" model has the largest coefficient magnitude, while the "between" model has the smallest coefficient magnitude.

Exercise 4 Understanding Fixed Effects

4.1

The following R chunk contains codes for writing and optimizing the log likelihood associated to the problem. The log likelihood, however, provides different answers depending on the initialized parameter values, suggesting a non-convering likelihood.

I have investigated into the matter of non-convergence individually and collectively, with classmates and a TA, to fix the problem. But, I unfortunately was not able to fix it.

```
#Randomly Selecting 100 Indices
ind100 = sample(x = 1:uniq_ID, size = 100, replace = FALSE)

#Extracting Dataframe Associated with 100 Indices
data_ex4 = data %>% filter(PERSONID %in% ind100)
```

```r
#Defining X and Y
x_ex4 = data_ex4 %>% select(EDUC, POTEXPER)
y_ex4 = data_ex4 %>% select(LOGWAGE)

#Initializing Intercepts
alphas = rep(0, 100)
init_intercept = list()

freq = rep(NA, 100)

for(i in 1:100){

  freq[i] = data %>% filter(PERSONID == ind100[i]) %>% nrow()
  init_intercept[[i]] = rep(alphas[i], freq[i])

}

init_intercept = init_intercept %>% unlist()

#Writing ML
normal_ML = function(parm){

  #Converting Y into Matrix
  Y = y_ex4 %>% as.matrix()

  #Adding Intercept Column
  X = x_ex4 %>% as.matrix()

  #Calculating XB
  XB = parm[1:100] + X %*% parm[101:102]

  #Normalizing the Quantity
  normal_Y = (Y - XB) / parm[103]

  #Likelihood
  lik = prod( dnorm(normal_Y) )

  #Returning the Likelihood
  return(lik)

}

#Writing Log ML
normal_logML = function(parm){

  #Converting Y into Matrix
  Y = y_ex4 %>% as.matrix()

  #Adding Intercept Column
  X = x_ex4 %>% as.matrix()

  #Initializing Intercepts
  init_intercept = list()
```

```
  freq = rep(NA, 100)

  for(i in 1:100){

    freq[i] = data %>% filter(PERSONID == ind100[i]) %>% nrow()
    init_intercept[[i]] = rep(parm[i], freq[i])

  }

  init_intercept = init_intercept %>% unlist()

  #Calculating XB
  XB = init_intercept + X %*% parm[101:102]

  #Normalizing the Quantity
  normal_Y = (Y - XB) / parm[103]

  #Calculating Log Likelihood
  log_lik =
    sum( dnorm(normal_Y, log = TRUE) )

  #Returning the Negative of the Log Likelihood
  return(-log_lik)

}


#Optimizing Log Likelihood
parm = rnorm(103)
opt = optim(par = parm, fn = normal_logML)
kable(data.frame(unique(data_ex4$PERSONID)[1:10], opt$par[1:10]),
      digits = 4,
      col.names = c("PERSONID","Fixed Effect"),
      caption = "Table of Individual Fixed Effect Parameter")
```

Table 4: Table of Individual Fixed Effect Parameter

PERSONID	Fixed Effect
29	-0.7884
50	-0.0480
124	0.2570
135	0.6251
151	-0.1657
179	-2.2170
212	-1.2567
232	0.3656
256	-0.0041
272	-0.9337

Comment

The table only presents the first 10 fixed-effect coefficients. I thought of presenting all the coefficients in the table, but I noticed that doing such will yield a table of coefficients that is three pages long. To look at the rest of fixed-effect coefficients, please run the code opt$par.

4.2

Here, we are running a regression to estimate the individual fixed effects on the invariant variables. Since the log likelihood did not converge in the previous problem, we run OLS to estimate the estimate the individual fixed effects first.

```
fixed_effects = lm(LOGWAGE ~ -1 + factor(PERSONID) + EDUC + POTEXPER, data = data_ex4) %>%
  coefficients() %>%
  .[1:100]

#Creating Dataframe for the Regression
data_ex4.2 = data.frame(

  fixed_effects,
  data_ex4 %>% select(PERSONID, ABILITY) %>% unique(),
  data_ex4 %>% select(PERSONID, MOTHERED) %>% unique(),
  data_ex4 %>% select(PERSONID, FATHERED) %>% unique(),
  data_ex4 %>% select(PERSONID, BRKNHOME) %>% unique(),
  data_ex4 %>% select(PERSONID, SIBLINGS) %>% unique()

)
colnames(data_ex4.2)[1] = "Fixed_Effects"

#Calculating the Coefficients
fixed_model = lm(Fixed_Effects ~  ABILITY +
                   MOTHERED + FATHERED +
                   BRKNHOME + SIBLINGS,
                 data = data_ex4.2)

fixed_coef = fixed_model %>%
  coefficients()

#Presenting Coefficients
kable(fixed_coef, digits = 4, col.names = "Coefficients",
      caption = "Table of Coefficients")
```

Table 5: Table of Coefficients

	Coefficients
(Intercept)	-0.6147
ABILITY	0.0354
MOTHERED	-0.0128
FATHERED	0.0129
BRKNHOME	0.1393
SIBLINGS	-0.0076

4.3

The standard errors in the previous problem may not be correctly estimated because of measurement errors. When we regress the fixed effects $\hat{\alpha}_{1:100}$ on the invariant variables in the previous problem, we implicitly assume that the $\hat{\alpha}$'s are without measurement errors. This, however, is a false statement because when the fixed effects are estimated in the first regression in the first part of exercise 4, it is estimated with measurement errors. For this reason, we opt to bootstrap to correctly estimate the standard errors.

```
iter = 49
beta_matrix = matrix(NA, nrow = iter, ncol = 6)

for(i in 1:iter){

  #Resampling Indices
  samp = sample(x = ind100, size = 100, replace = TRUE)

  #Sampling Individuals with Resampled Indices
  boot_df_samp = data.frame()

  for(j in 1:100){

    boot_df_samp = rbind( boot_df_samp, filter(data, PERSONID == samp[j]) )

  }

  #Estimating Fixed Effects (Alphas)
  boot_fixed_effects =
    lm(LOGWAGE ~ -1 + EDUC + POTEXPER + factor(PERSONID), data = boot_df_samp) %>%
    coefficients() %>%
    .[-c(1, 2)]

  #Subsetting Invariant Variables
  boot_invariant = boot_df_samp %>%
    group_by(PERSONID) %>%
    select(PERSONID, ABILITY, MOTHERED, FATHERED, BRKNHOME, SIBLINGS) %>%
    unique() %>%
    arrange(PERSONID)

  #Creating Dataframe for Second Regression
  boot_df = data.frame(

    PERSONID = boot_invariant$PERSONID,
    fixed_effects = boot_fixed_effects,
    ABILITY = boot_invariant$ABILITY,
    MOTHERED = boot_invariant$MOTHERED,
    FATHERED = boot_invariant$FATHERED,
    BRNKHOME = boot_invariant$BRKNHOME,
    SIBLINGS = boot_invariant$SIBLINGS

  )

  #Regressing Fixed Effects on Invariant Variables
  beta_matrix[i, ] = lm(fixed_effects ~ ABILITY + MOTHERED + FATHERED + BRNKHOME + SIBLINGS,
                 data = boot_df) %>%
```

```
    coefficients()


}

#Calculating Bootstrap Standard Errors
boot_sd = apply(beta_matrix, 2, sd) %>% as.data.frame()
colnames(boot_sd) = "Bootstrap SE"
rownames(boot_sd) = c("Intercept", "ABILITY", "MOTHERED", "FATHERED", "BRNKHOME", "SIBLINGS")

#Presenting the Bootstrap Standard Errors
kable(boot_sd, digits = 4,
      caption = "Table of Bootstrap Standard Errors")
```

Table 6: Table of Bootstrap Standard Errors

	Bootstrap SE
Intercept	0.3858
ABILITY	0.0422
MOTHERED	0.0131
FATHERED	0.0089
BRNKHOME	0.0937
SIBLINGS	0.0203