# ECON-613 HW #4

*Peter Kim*

*2019-04-13*

## 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()
## )
```

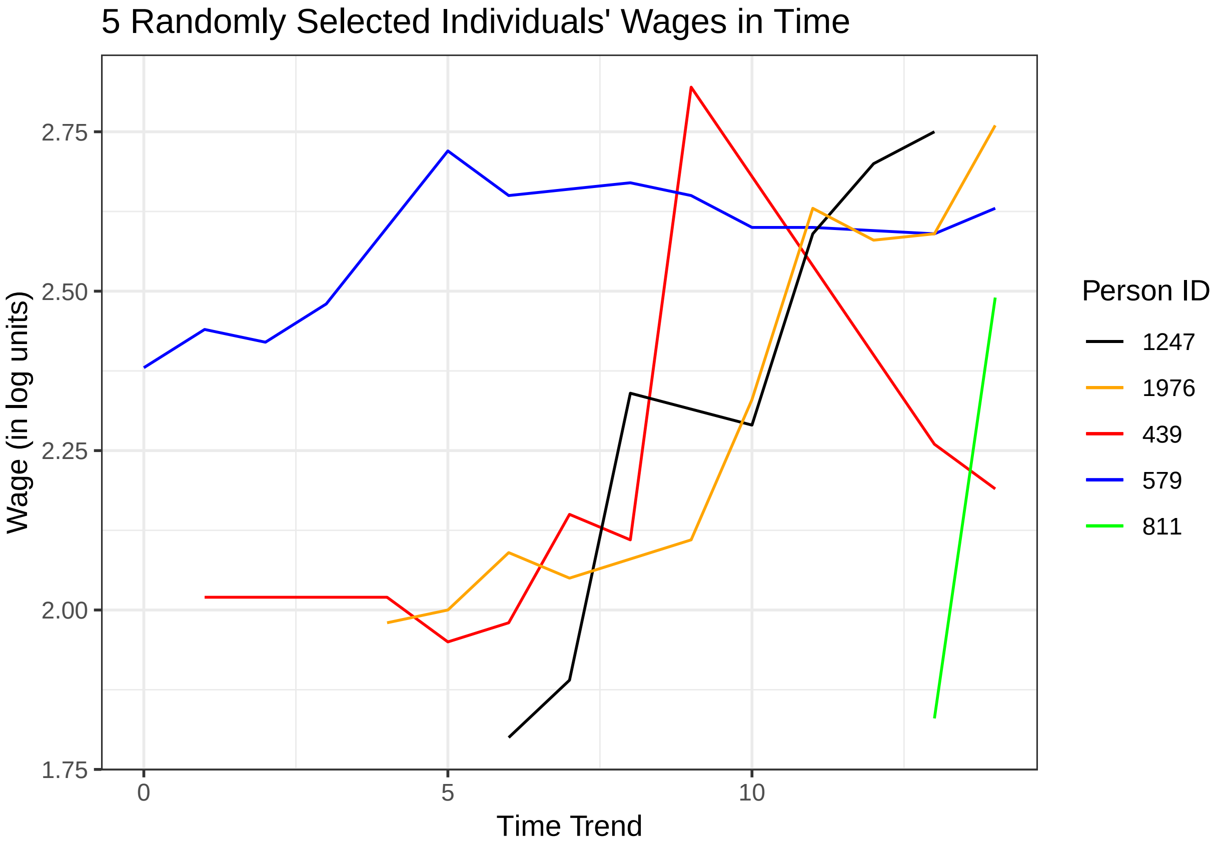
```
#Identifying Unique Person ID
uniq_ID = unique(data$PERSONID) %>% length()

#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 in 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 =
```

```r
  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 $\hat{\beta}_{POTEXPER}$

```r
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.

```r
#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 individual fixed effects on the invariant variables.

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

  opt$par[1:100],
  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] = "Opt"

#Calculating the Coefficients
fixed_model = lm(Opt ~  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) | 0.0472 |
| ABILITY | 0.1150 |
| MOTHERED | 0.0305 |
| FATHERED | -0.0251 |
| BRKNHOME | -0.1381 |
| SIBLINGS | -0.0491 |

### 4.3

The standard errors in the previous problem may not be correctly estimated because of a potentital heteroskedasticity in the model - panel data may suffer from a non-constant variance with respect to time. Fortunately, this issue can be fixed by calculating heteroskedasticity-consistent standard errors (or Eicker–

Huber–White standard errors). In the following, I use the package “ivpack” to calculate the Huber-White standard errors.

```
#Installing the Package ivpack
install.packages("ivpack")
```

```
## Installing package into '/home/grad/MK408/R/x86_64-redhat-linux-gnu-library/3.5'
## (as 'lib' is unspecified)
```

```
#Fitting Fixed Effects Model in Exericse 4
ex_4_model = lm(LOGWAGE ~ - 1 + factor(PERSONID) + EDUC + POTEXPER, data = data_ex4)

#Calculating Huber-White Standard Errors
HW_sd = ivpack::robust.se(ex_4_model)[ c("EDUC", "POTEXPER") , "Std. Error"] %>% as.data.frame()
```

```
## [1] "Robust Standard Errors"
```

```
kable(HW_sd, digits = 4, col.names = "Standard Error",
      caption = "Table of Huber-White Standard Errors")
```

Table 6: Table of Huber-White Standard Errors

| | Standard Error |
|---|---|
| EDUC | 0.0254 |
| POTEXPER | 0.0040 |