HW4

Jie Ren

April 4, 2019

Exercise 1

```
rm(list=ls())
setwd("C:/Users/jiere/Dropbox/Spring 2019/ECON 613/ECON613_HW/HW4_output")
# install.packages("lme4")
# install.packages("Matrix")
# install.packages("ggplot2")
# install.packages("reshape2")
library("reshape2")
library("ggplot2")
library("lme4")
kt <- read.csv("Koop-Tobias.csv")
```

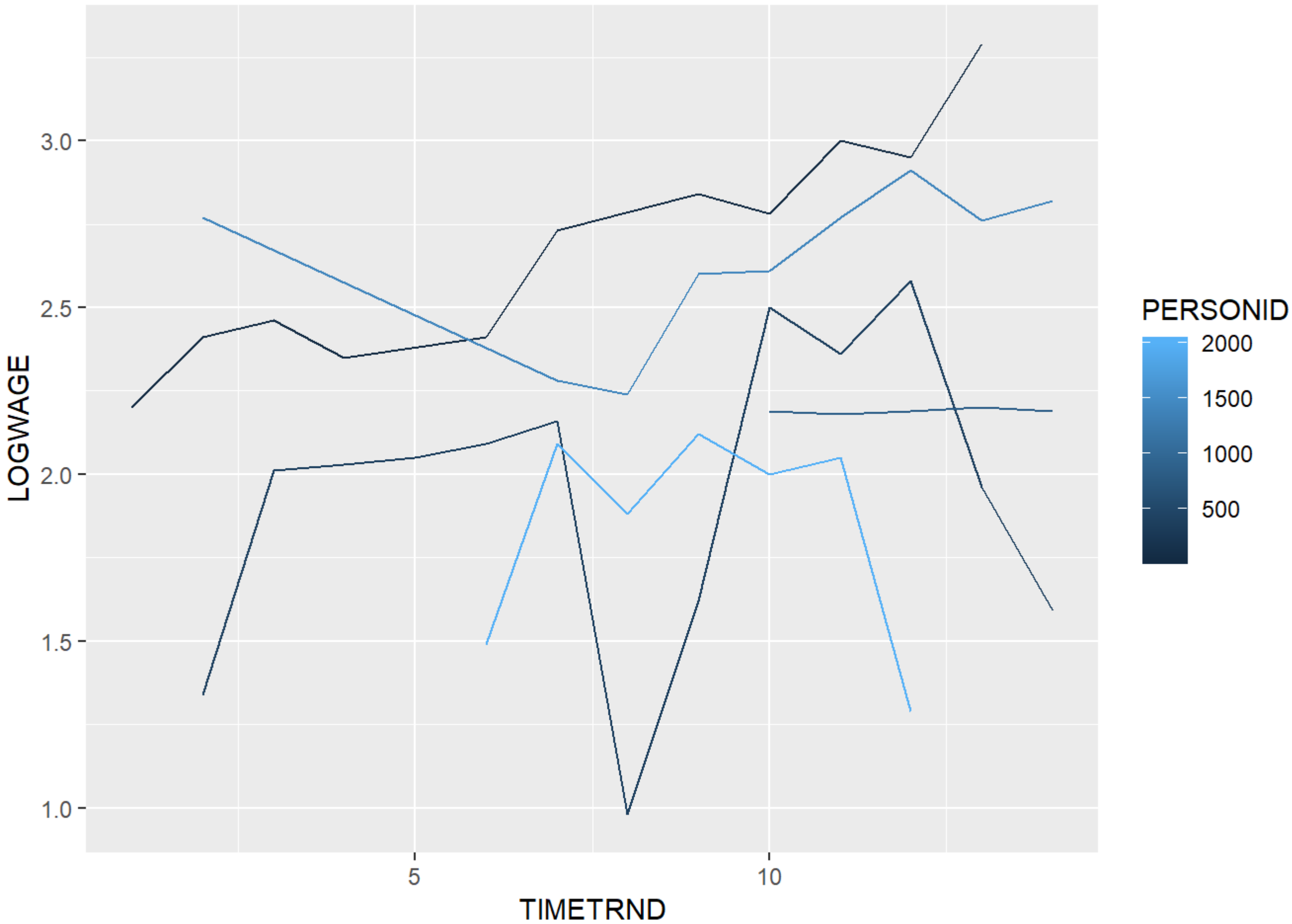
Find the sample dimension for 5 randomly selected individual

```
rd <- sample(1:2178,5)
dimension <- aggregate(list(Dimension = kt$PERSONID), list(PERSONID = kt$PERSONID), length)[rd,]
rownames(dimension) <- NULL
dimension
```

```
##   PERSONID Dimension
## 1     1282         1
## 2     1023         5
## 3      791         7
## 4     1808         2
## 5     1548         4
```

```
rnd <- sample(1:2178,5)
kt.sub <- kt[kt$PERSONID %in% rnd,c("PERSONID","LOGWAGE","TIMETRND")]

ggplot(kt.sub, aes( x=TIMETRND, y=LOGWAGE, group=PERSONID, col=PERSONID)) +
  geom_line()
```



Noticed that this is an unbalanced panel and the time trend variable is not consecutive

Exercise 2

Check with lme4 package

```
re.lm <- lmer(LOGWAGE ~ EDUC + POTEXPER + (1|PERSONID), data = kt)
summary(re.lm)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: LOGWAGE ~ EDUC + POTEXPER + (1 | PERSONID)
##    Data: kt
## 
## REML criterion at convergence: 16700.7
## 
## Scaled residuals: 
##     Min      1Q  Median      3Q     Max 
## -7.1639 -0.4559  0.0635  0.5351  7.3176 
## 
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  PERSONID (Intercept) 0.1330   0.3647  
##  Residual             0.1129   0.3360  
## Number of obs: 17919, groups:  PERSONID, 2178
## 
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept) 0.5667942  0.0434683   13.04
## EDUC        0.1077081  0.0033500   32.15
## POTEXPER    0.0387584  0.0007186   53.93
## 
## Correlation of Fixed Effects:
##          (Intr) EDUC  
## EDUC     -0.972       
## POTEXPER -0.066 -0.070
```

Exercise 2&3

Calculate the random effect using the transformed model (Mannually)

(Method from Principle of Econometrics) Fisrt let's define a function for between estimator

```
tmean <- function(x,id,rep = T){# calculate the mean for each individual id, and and repeat thes
e mean for the same id
  # or alternatively use ave
  dim     <- aggregate(list(Tp = id), list(id = id), length)
  mean    <- aggregate(list(idmean = x),list(id = id),mean)
  gpmean  <- rep(mean$idmean,dim$Tp)
  ifelse(rep == T, return(gpmean),return(mean$idmean))
}


fix.bt <- function(y,X,id){
  dep     <- tmean(y,id,rep = F)
  indep   <- apply(X,2,tmean, id = id, rep = F)
  result  <- lm(dep~indep)
  return(result)
}
```

The let's define a funciton for with-in estimator

```r
fix.wi <- function(y,X,id){
  dep     <- y - tmean(y,id)
  indep   <- as.matrix(X - apply(X,2,tmean, id = id))
  result  <- lm(dep~0+indep)
  return(result)
}
```

Using the variance from these two estimator, we are able the caculate the vairance of residual in random effect estimator. Then we can get get the transformed model and simply do OLS!

```r
fix.re.ib <- function(y,X,id){
  N <- length(id)
  n <- length(unique(id))
  k <- ncol(X)
  sigma2_u <- sum((summary(fix.bt(y,X,id))$residual)^2)/(n-k)
  sigma2_e <- sum((summary(fix.wi(y,X,id))$residual)^2)/((N-n)-k)

  # for unbalanced panel use harmonized mean of time period of each id has
  dim       <- aggregate(list(Tp = id), list(id = id), length)
  Th        <- length(unique(id))/sum(1/dim$Tp)
  sigma2_v  <- sigma2_u - sigma2_e/Th

  # Calculate lumbda for the tranformed model
  Tp_i    <- rep(dim$Tp,dim$Tp)
  lambda  <- 1-sqrt(sigma2_e/(Tp_i*sigma2_v + sigma2_e)) # special case of unbalanced panel

  # Transformed model
  X       <- cbind(Intercept = 1,X)
  dep     <- y - lambda*tmean(y,id)
  indep   <- as.matrix(X - rep(lambda,ncol(X))*apply(X,2,tmean, id = id))
  result  <- lm(dep~0+indep)
  return(result)
}
```

First time difference estimator

Here we regard the discouninous timetime trend as continous, per Professor Sidibe.

```r
fix.fd <- function(y,X,id){
  df <- data.frame(y,X,id)
  for (i in 1:ncol(X)){
    df <- transform(df, col=ave(df[,i+1], df$id, FUN = function(x) c(NA, diff(x)))) # MUST inclu
de FUN, or cause error
    names(df)[ncol(df)]<-paste("indep",i,sep=" ")
  }
  df <- transform(df, dep=ave(y, id, FUN = function(x) c(NA, diff(x))))

  indep   <- as.matrix(na.omit(df[,grepl("indep",colnames(df))]))
  dep     <- na.omit(df[,"dep"])
  result  <- lm(dep~0+indep)
  return(result)
}

coef(fix.re.ib(kt$LOGWAGE,kt[,c("EDUC","POTEXPER")],kt$PERSONID)) # random
```

```
## indepIntercept      indepEDUC  indepPOTEXPER
##     0.56356104     0.10793517     0.03876441
```

```r
coef(fix.wi(kt$LOGWAGE,kt[,c("EDUC","POTEXPER")],kt$PERSONID)) # with-in
```

```
##     indepEDUC indepPOTEXPER
##    0.12366202    0.03856107
```

```r
coef(fix.bt(kt$LOGWAGE,kt[,c("EDUC","POTEXPER")],kt$PERSONID)) # between
```

```
##   (Intercept)     indepEDUC indepPOTEXPER
##    0.84556883    0.09309987    0.02599874
```

```r
coef(fix.fd(kt$LOGWAGE,kt[,c("EDUC","POTEXPER")],kt$PERSONID)) # first difference
```

```
## indepindep.1 indepindep.2
##   0.04793559   0.03286052
```

Their result are close for random effect and with-in estimator, but others are quite different

Exercise 4

```
# randomly selection 100 individual
rnd <- sample(1:2178,100)
kt.rand <- kt[kt$PERSONID %in% rnd,]

fix.dvls <- function(df, y_name, X_name,id_name,dmatrix = F){
  idcol <- which( colnames(df)== id_name )
  idx   <- unique(df[,idcol])
  for (i in 1:100){
    df$D <- 0
    df[df[,idcol] == idx[i],ncol(df)] <- 1
    names(df)[ncol(df)] <- paste("Dummy_",idx[i],sep = "")
  }
  dummy <- df[,grepl("Dummy_",colnames(df))]
  dummy <- dummy[,-1] # drop the first person to avoid dummy variable trap
  print(paste("use the first selected individual as reference, which id =",idx[1],sep = " "))

  dep       <- df[,y_name]
  indep     <- as.matrix(cbind(df[,X_name],dummy))
  result.d  <- lm(dep~0+indep)
  ifelse(dmatrix == F,return(result.d),return(dummy))
}

individual <- coef(fix.dvls(kt.rand,"LOGWAGE",c("EDUC","POTEXPER"),"PERSONID"))
```

```
## [1] "use the first selected individual as reference, which id = 32"
```

```
individual <- individual[3:length(individual)]



# do with MLE
dummy <- fix.dvls(kt.rand,"LOGWAGE",c("EDUC","POTEXPER"),"PERSONID",dmatrix = T)
```

```
## [1] "use the first selected individual as reference, which id = 32"
```

```
# Define likelihood function
ll.DVLS <- function(b){
  n    <- length(y)
  b    <- b[2:length(b)]
  sig2 <- b[1]
  ll   <- -n/2*log(2*pi)-n/2*log(sig2)-(y-X%*%b)^2/(2*sig2)
  ll.s <- -sum(ll)
  ll.s
  return(ll.s)
}
set.seed(1)
y <- kt.rand$LOGWAGE
X <- cbind(as.matrix(kt.rand[,c("EDUC","POTEXPER")]),as.matrix(dummy))
b <- rnorm(102)

result <- optim(b, ll.DVLS)
result$par
```

```
##   [1] -0.626453811  0.183643324 -0.595466836  1.595280802  0.329507772
##   [6] -0.820468384  0.487429052  0.738324705  0.575781352 -0.305388387
##  [11]  1.511781168  0.389843236 -0.621240581 -2.214699887  1.124930918
##  [16] -0.044933609 -0.016190263  0.943836211  0.821221195  0.593901321
##  [21]  0.918977372  0.782136301  0.074564983 -1.989351696  0.619825748
##  [26] -0.056128740 -0.155795507 -1.470752384 -0.478150055  0.417941560
##  [31]  1.358679552 -0.102787727  0.387671612 -0.053805041 -1.377059557
##  [36] -0.414994563 -0.394289954 -0.059313397  1.100025372  0.763175748
##  [41] -0.164523596 -0.253361680  0.696963375  0.556663199 -0.688755695
##  [46] -0.707495157  0.364581962  0.768532925 -0.112346212  0.881107726
##  [51]  0.398105880 -0.612026393  0.341119691 -1.129363096  1.433023702
##  [56]  1.980399899 -0.367221476 -1.044134626  0.569719627 -0.135054604
##  [61]  2.401617761 -0.039240003  0.689739362  0.028002159 -0.743273209
##  [66]  0.188792300 -1.804958629  1.465554862  0.153253338  2.172611670
##  [71]  0.475509529 -0.709946431  0.610726353 -0.934097632 -1.253633400
##  [76]  0.291446236 -0.443291873  0.001105352  0.074341324 -0.589520946
##  [81] -0.568668733 -0.135178615  1.178086997 -1.523566800  0.593946188
##  [86]  0.332950371  1.063099837 -0.304183924  0.370018810  0.267098791
##  [91] -0.542520031  1.207867806  1.160402616  0.700213650  1.586833455
##  [96]  0.558486426 -1.276592208 -0.573265414 -1.224612615 -0.473400636
## [101] -0.620366677  0.042115873
```

NOTE: this likelihood funcion is not converging.

Regressing using the time invarient variables

```
kt.rand.tiv <- kt.rand[!duplicated(kt.rand$PERSONID),] # keep one obs for each person
kt.rand.tiv <- kt.rand.tiv[-1,]
kt.rand.tiv$individual <- individual
result.2 <- lm(individual~ABILITY + MOTHERED + FATHERED + BRKNHOME + SIBLINGS, kt.rand.tiv)
summary(result.2)
```

```
##
## Call:
## lm(formula = individual ~ ABILITY + MOTHERED + FATHERED + BRKNHOME +
##     SIBLINGS, data = kt.rand.tiv)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.10539 -0.17659  0.05542  0.22772  0.82684
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.658569   0.210630   3.127  0.00236 **
## ABILITY      0.039573   0.043422   0.911  0.36447
## MOTHERED    -0.007443   0.017764  -0.419  0.67619
## FATHERED     0.008709   0.013196   0.660  0.51092
## BRKNHOME     0.157447   0.104249   1.510  0.13436
## SIBLINGS     0.018253   0.018000   1.014  0.31318
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3953 on 93 degrees of freedom
## Multiple R-squared:  0.04502,    Adjusted R-squared:  -0.00632
## F-statistic: 0.8769 on 5 and 93 DF,  p-value: 0.4998
```

For with-in estimator: Since there may be the issue of heteroskedasticity causing by the correlation of error term between each individual, a sandwich form (huber white) standard error is needed rathe than standard ols se.

```
X <- as.matrix(kt[,c("EDUC","POTEXPER")])
inv_XX <- solve(t(X) %*% X)
residual <- fix.wi(kt$LOGWAGE,kt[,c("EDUC","POTEXPER")],kt$PERSONID)$residuals

D <- t(X) %*% diag(residual)^2 %*% X

EHW <- inv_XX %*% D %*% inv_XX

diag(sqrt(EHW))
```

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