> # Jeff Hill, ECON 613 Assignment 4

> library(plyr)

> library(dplyr)

> library(reshape2)

> library(fastDummies)

> library(nlme)

> library(plm)

> library(nnet)

> # set seed

> set.seed(613)

>

> # load data

> datapath <- "/Users/admin/Documents/Econ\_613/Data/Assignment 4/"

> koop <- read.csv(file=paste(datapath,"Koop-Tobias.csv",sep=""), header=TRUE, sep=",")

>

> #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

> # Exercise 1 Data

> #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

> # randomly select 5 of the 2178 individuals

> sample <- subset(koop[,c(1,3,5)], subset = PERSONID %in% c(sample(1:2178,5,replace=F)))

> # with seed = 613, the individuals should be: 226, 707, 1790, 2081, and 2132

>

> # since our sample df is in long form, convert it to wide form for clarity, and rename it wagepanel

> wagepanel <- dcast(sample, PERSONID ~ TIMETRND, value.var="LOGWAGE")

>

> # wagepanel is missing a column for time period 1, as no individuals have data for that period.

> # add in that column and then rename rows and columns

> wagepanel <- wagepanel[,c(2,1,3:15)]

> colnames(wagepanel) <- c('t0','t1','t2','t3','t4','t5','t6','t7','t8','t9','t10','t11','t12','t13','t14')

> rownames(wagepanel) <- c(226,707,1790,2081,2132)

> wagepanel$t1 <- c(NA,NA,NA,NA,NA)

> # wagepanel now contains the panel dimension of wages for 5 randomly selected individuals.

> wagepanel

t0 t1 t2 t3 t4 t5 t6 t7 t8 t9 t10 t11 t12 t13 t14

226 2.1 NA 2.39 2.53 2.52 NA 1.87 2.01 2.76 2.76 2.75 2.84 2.70 2.64 2.66

707 NA NA NA 1.67 1.79 2.31 1.92 NA 2.28 NA 1.87 1.58 2.38 2.41 2.37

1790 NA NA 2.88 2.88 2.98 2.94 2.94 3.02 NA 3.14 3.24 3.20 3.16 3.18 3.14

2081 NA NA 1.81 2.10 2.03 1.48 NA NA NA NA NA NA NA NA NA

2132 NA NA NA NA NA NA NA 1.93 1.87 1.95 1.95 2.18 2.21 2.16 2.49

>

>

>

> #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

> # Exercise 2 Random Effects

> #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

>

> # Random effects model using gls

> REfit <- gls(LOGWAGE ~ EDUC + POTEXPER, data=koop)

>

> # notice that the results from this regression yield the same coefficients as a linear OLS

> # however the standard errors may differ.

>

>

> #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

> # Exercise 3 Fixed Effects Model

> #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

>

>

> # Between Estimator------------------------------

>

> # turn our koop-tobias data from long to wide for the sake of the between estimator but also because

> # wide data is easier to manipulate in certain regards

> koopwide <- recast(koop, PERSONID ~ TIMETRND + variable, measure.var = c("LOGWAGE", "POTEXPER","EDUC"))

> # reorder columns

> koopwide <- koopwide[,c(1,2,5,8,11,14,17,20,23,26,29,32,35,38,41,44,

+ 3,6,9,12,15,18,21,24,27,30,33,36,39,42,45,

+ 4,7,10,13,16,19,22,25,28,31,34,37,40,43,46)]

>

> # define which columns are LOGWAGE, POTEXPER, and EDUC for convenience:

> lwagecol <- c(2:16)

> pexpcol <- c(17:31)

> educcol <- c(32:46)

> # generate averages for logwage and potential experience across time for each individual.

> koopwide$lwageavg <- rowMeans(koopwide[,lwagecol], na.rm = TRUE)

> koopwide$pexpavg <- rowMeans(koopwide[,pexpcol], na.rm = TRUE)

> koopwide$educavg <- rowMeans(koopwide[,educcol], na.rm = TRUE)

>

> # Now with the data in this wide format, we can just run a linear model on these average values to

> # calculate the between estimator

> betweenfit <- lm(lwageavg ~ educavg + pexpavg, koopwide)

> betweenfit$coefficients

(Intercept) educavg pexpavg

0.84556883 0.09309987 0.02599874

>

>

> # Within Estimator------------------------------

>

> # Now we will need to merge our averages for lwage, potexper, and educ we calculated for the between

> # estimator back into our long dataframe (namely koop), since we need data for each individual in

> # each time period for the within estimator

> koop <- merge(koop,koopwide[,c("PERSONID","lwageavg","pexpavg","educavg")], by.x = "PERSONID", by.y = "PERSONID")

>

> # now that we have the average lwage, education, and potential experience for each individual, we need to generate

> # the differences from the means for each individual in each time period

> koop$lwagewithin <- koop$LOGWAGE - koop$lwageavg

> koop$pexpwithin <- koop$POTEXPER - koop$pexpavg

> koop$educwithin <- koop$EDUC - koop$educavg

>

> # run the model

> withinfit <- lm(lwagewithin ~ educwithin + pexpwithin -1, koop)

> withinfit$coefficients

educwithin pexpwithin

0.12366202 0.03856107

>

>

> # First Difference Estimator --------------------

>

> # I subset the columns of interest in koops into a dataframe called firstd which i will manipulate for the first

> # difference estimator. Also of note is that the time periods missing are not systematic. Looking at individual 1

> # for example, they have data for time periods 0,7,9, and 10. For our first difference model, we only consider

> # adjacent time periods, and we accomplish that using trend\_diff below

>

> firstd <- koop[,c("PERSONID","LOGWAGE","EDUC","POTEXPER","TIMETRND")]

> # create lagged columns for lwage, educ, and potexper that yield NAs for the first time period of each individual

> # we then take the difference between periods and place the difference values in columns lwagelag, educlag, etc.

> # this is all done in the commands below

> firstd$lwagelag <- unlist(by(firstd$LOGWAGE , list(firstd$PERSONID) , function(i) c(NA,diff(i))))

> firstd$educlag <- unlist(by(firstd$EDUC , list(firstd$PERSONID) , function(i) c(NA,diff(i))))

> firstd$pexplag <- unlist(by(firstd$POTEXPER , list(firstd$PERSONID) , function(i) c(NA,diff(i))))

> firstd$trend\_diff <- unlist(by(firstd$TIMETRND, list(firstd$PERSONID), function(i) c(NA,diff(i))))

> # this last column now calculates the difference between time periods. we only want to regress on data where

> # the time difference is 1.

>

>

> # run the model

> firstdfit <- lm(lwagelag ~ educlag + pexplag -1, subset(firstd, trend\_diff == 1))

> firstdfit$coefficients

educlag pexplag

0.04310838 0.05353695

>

>

> # Create a table for comparison of coefficients ------------

>

> comp\_df <- data.frame(matrix(NA, ncol = 3, nrow = 3))

> colnames(comp\_df) <- c("Between","Within","First Difference")

> rownames(comp\_df) <- c("Intercept","Education","Experience")

>

> comp\_df[,1] <- betweenfit$coefficients

> comp\_df[,2] <- c(NA,withinfit$coefficients)

> comp\_df[,3] <- c(NA,firstdfit$coefficients)

>

> # comp\_df now is a dataframe comparing the coefficents of the 3 estimators

> comp\_df

Between Within First Difference

Intercept 0.84556883 NA NA

Education 0.09309987 0.12366202 0.04310838

Experience 0.02599874 0.03856107 0.05353695

>

> #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

> # Exercise 4 Understanding Fixed Effects

> #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

> # subset our koop dataframe to randomly select 100 individuals

> FEkoop <- subset(koop, subset = PERSONID %in% c(sample(1:2178,100,replace=F)))

>

> # betas contain our two betas for education and experience, and then 100 alphas.

>

> # subset our x data to just pull the columns we want

> FEx <- FEkoop[,c(2,4,1)]

> FEy <- FEkoop[,3]

>

> test <- as.matrix(FEx[,c(1,2)])%\*%c(3,3)

>

> FE\_neg\_ll <- function (param,x,y) {

+ # x will contain education and experience, as well as idividual id since we need to match the alphas to

+ # the individuals

+ # parameters will be vector of length 103. the first 100 will be the 100 alphas, parameters 101 and 102 will

+ # be the two betas for educ and exper, and parameter 103 will be the sigma

+ sigma <- param[103]

+ alpha <- param[1:100]

+

+ # i need to map the alphas for each individual to the observations for each individual. I do that below

+ mapdf <- data.frame(old=c(x[with(x, c(PERSONID[-1]!= PERSONID[-nrow(x)], TRUE)),3]),new=param[1:100]) # this creates a map between unique values of person ID

+ # (which we have 100 of) and the 100 alphas (just 1 to 100)

+ alphalong <- mapdf$new[match(FEx[,3],mapdf$old)] # this maps the alphas to the person IDs, and creates the

+ # alpha vector which is of correct length, namely 798, which matches the dim of our X matrix.

+

+ cut\_x <- x[,c(1,2)] # dropping the person ID column of x since it was only used for matching the alphas

+ xb <- as.matrix(cut\_x)%\*%param[101:102] + alphalong # create XB

+ u <- y - xb # create u, the difference between y and yhat

+ -sum(log(dnorm(u,sd = sigma))) # sum the log of each individual likelihood, then make negative

+ }

>

> FEfit <- nlm(FE\_neg\_ll,rep(1,103),x=FEx,y=FEy)

There were 50 or more warnings (use warnings() to see the first 50)

>

> # this optimization is very sensitive to starting values. with starting values that are 0 or negative, the optimization

> # yields incorrect results (significantly higher likelihood). with starting values above 5, the optimization yields

> # incorrect results (significiantly higher likelihood). With starting values between .1 and 5, the optimization yields the

> # lowest likelihood (lik minimum = 145.3604). Since the estimates are consistent across this range, and higher or lower

> # starting values we get less optimal likelihood, I will use the results from this range as my estimates. also note

> # that I varied the 100 alphas, 2 betas, and the standard deviation separately.

>

>

>

> alphas <- FEfit$estimate[1:100]

> betas <- FEfit$estimate[101:102]

> sigma <- FEfit$estimate[103]

>

> # parameters reported below:

> alphas

[1] 0.376954219 -0.290885226 0.757820203 -0.298875834 0.645238908 0.564340045 0.583091251 0.481507880 0.402466056 0.817543435

[11] 0.713466216 1.022051826 0.378734509 0.399410440 0.713575967 0.364344612 0.580368209 0.832189139 0.299499935 0.551012373

[21] 0.564101111 0.796191776 0.702625256 0.550827089 0.687741517 1.004361783 0.165546858 0.470209189 0.743339319 -0.167274026

[31] 0.241068825 0.556698968 0.506730931 0.680943252 1.143515979 0.276926472 -0.251084341 1.171574657 1.013577422 0.895535370

[41] 0.778986961 0.781925680 -0.053838659 0.875986470 -0.354226345 0.714226855 0.598644954 0.918148464 0.937227927 0.720550472

[51] 1.506599558 -0.086898498 0.045461499 0.953860315 -0.201253641 0.491011629 0.406158931 0.655148059 0.447587707 0.566681013

[61] 0.638373708 0.477782524 0.977669390 0.005081855 0.069972027 0.472953598 0.121734130 0.379928527 0.063146443 0.415721746

[71] 0.802020999 0.360025118 0.500817275 0.657512095 0.947368801 0.589415457 0.568262618 0.328954516 0.598815687 0.651749124

[81] 0.270509370 0.522613055 0.553541866 1.108699389 -0.020060440 -0.003125984 0.012283488 0.411731705 1.290729557 -0.028032121

[91] 0.178843335 -0.138607491 0.212787537 0.348790080 0.894571164 0.379419043 1.101281703 0.634498739 0.048620785 0.242472181

> betas

[1] 0.11275066 0.04081465

> sigma

[1] 0.2902956

>

> # subset our FE data to get the time-invariant data for our 100 individuals, and add alphas to the data.

> FE\_time\_inv <- unique(FEkoop[,c(6:10)])

> FE\_time\_inv$alpha <- alphas

>

> # here we regress the alphas on the time invariant variables.

> time\_inv\_fit <- lm(alpha ~ ABILITY + MOTHERED + FATHERED + BRKNHOME + SIBLINGS, FE\_time\_inv)

>

>

> # The standard errors are potentially not correctly estimated, because the alphas are generated from our optimization,

> # and so they carry some amount of error in their estimation. But we then treat them as our endogenous variable in the regression,

> # not accounting for the error in their values. so to correct for these standard errors, we bootstrap.

>

>

>

> # FEkoop going in

>

> fe\_bootstrap <- function(data, reps) {

+ beta\_df <- data.frame(matrix(NA, nrow = reps, ncol = 6)) # here we create the empty matrix to temporarily store betas in

+ colnames(beta\_df) <- c("Intercept", "Ability","MotherEd","FatherEd","Broken Home","Siblings") # name the columns

+ for (i in 1:reps) {

+ ind\_list <- sample <-sample(unique(data$PERSONID), 100, replace = T) # sample a list of the 100 individuals selected with replacement

+ boot\_sample <- data.frame() # boot\_sample will be FEkoop resampled with replacement of individuals each iteration

+ for(j in ind\_list) {

+ boot\_sample <- rbind(boot\_sample,data[which(data$PERSONID==j),])

+ }

+ boot\_ss <- boot\_sample[,1:4] # take only the columns of interest from boot\_sample

+ boot\_ss <- with(boot\_ss, data.frame(class.ind(PERSONID), EDUC, POTEXPER, LOGWAGE)) # generates all the dummy variables for the model

+ boot\_fit <- lm(LOGWAGE~.,boot\_ss) # run the FE model with dummies using OLS

+ # using ols here vs the log-lik function for speed and becuase of the potential inconsistency of the Log-lik with different

+ # starting parameters.

+

+ alphas <- as.vector(boot\_fit$coefficients[2:(length(boot\_fit$coefficients)-2)]) # pull out all the alphas

+ boot\_time\_inv <- unique(boot\_sample[,c(6:10)]) # gather our time-invariant variables still in boot\_sample

+ boot\_time\_inv$alpha <- alphas

+ boot\_inv\_fit <- lm(alpha ~ ABILITY + MOTHERED + FATHERED + BRKNHOME + SIBLINGS, boot\_time\_inv) # run alpha ~ time invariants

+ beta\_df[i,] <- t(as.vector(boot\_inv\_fit$coefficients)) # store the betas

+ }

+ apply(beta\_df, 2, sd, na.rm = T)

+ }

> fe\_bootstrap(FEkoop,199)

Intercept Ability MotherEd FatherEd Broken Home Siblings

0.35212144 0.05707063 0.02190639 0.01899293 0.15895053 0.02352577

> # the end result is 5 standard errors (6 if you include constant) for our 5 time invariant variables on alpha generated through bootstrap.