knitr::opts\_chunk$set(echo = TRUE)  
#+ setup, error=TRUE

## Data Preparation

# Brian Kang  
# Must use 64-bit version of R  
#knitr::opts\_chunk$set(error=TRUE)  
#rm(list = ls())  
  
setwd("H:/Honors Research")  
#chooseCRANmirror(graphics=FALSE, ind=1)  
#install.packages(c("hdm","dplyr","stringr","glmnet","gdata","fastDummies"))  
library(hdm)  
library(dplyr)

library(stringr)  
library(glmnet)

library(gdata)

library(fastDummies)  
library(lmtest)

library(car)

# Import dataset --------------------------------------------------------------------  
temp <- read.csv("psam\_pusb.csv", header = T, nrows = 1)  
# columns that don't import as factors using default settings  
fnf <- c("DIVISION","PUMA","REGION","ST","ADJINC","CIT","COW","DDRS","DEAR",  
 "DEYE","DOUT","DPHY","DRAT","DRATX","DREM","ENG","FER","GCL","GCM","GCR",  
 "HINS1","HINS2","HINS3","HINS4","HINS5","HINS6","HINS7","JWTR","LANX",  
 "MAR","MARHD","MARHM","MARHT","MARHW","MIG","MIL","MLPA","MLPB","MLPCD",  
 "MLPE","MLPFG","MLPH","MLPI","MLPJ","MLPK","NWAB","NWAV","NWLA","NWLK",  
 "NWRE","RELP","SCH","SCHG","SCHL","SEX","WKL","WKW","WRK","ANC","ANC1P",  
 "ANC2P","DECADE","DIS","DRIVESP","ESP","ESR","FOD1P","FOD2P","HICOV",  
 "HISP","INDP","JWAP","JWDP","LANP","MIGPUMA","MIGSP","MSP","NATIVITY",  
 "NOP","OC","OCCP","PAOC","POBP","POWPUMA","POWSP","PRIVCOV","PUBCOV",  
 "QTRBIR","RAC1P","RAC2P","RAC3P","RACAIAN","RACASN","RACBLK","RACNH",  
 "RACNUM","RACPI","RACSOR","RACWHT","RC","SCIENGP","SCIENGRLP","SFN",  
 "SFR","VPS","WAOB")  
# columns that do import as factors using default setting  
fif <- c("RT","SERIALNO","NAICSP","SOCP")  
# all columns that are factors  
fcf <- append(fnf,fif)  
fcf <- append(fcf,names(temp[,c(131:286)]))  
# vector of classes of data columns  
colclass <- ifelse(colnames(temp) %in% fcf, 'factor', 'numeric')  
  
temp1 <- read.csv("psam\_pusa.csv", header = T, colClasses = colclass) # U.S. PUMS data  
temp2 <- read.csv("psam\_pusb.csv", header = T, colClasses = colclass)  
dat <- rbind(temp1,temp2)  
dat <- dat[,-c(1,2)] # drop unecessary IDs  
dat <- dat[,-c(129:284)] # drop unnecessary flag vars  
  
# US Census: "Income used to calculate poverty status includes PERNP (earnings) and PINCP (income)"  
  
# Calculate Income-Poverty ratio ----------------------------------------------------  
# (POVPIP only shows NA or <0.5 or >=0.5 so calculate actual ratio)  
PovertyThreshold <- rep(NA, nrow(dat))  
getThreshold <- function(threshold) { # values from CPS 2018  
 for (i in 1:nrow(dat)) {  
 if (dat$AGEP[i] < 18) { # under 18 yrs  
 threshold[i] <- NA  
 } else if (dat$SPORDER[i]==1 & dat$AGEP[i] < 65) { # individual  
 threshold[i] <- 13064  
 } else if (dat$SPORDER[i]==1 & dat$AGEP[i] >= 65) {  
 threshold[i] <- 12043  
 # AGEP doesn't have NA values  
 } else if (dat$SPORDER[i]==2 & dat$AGEP[i] < 65 & dat$OC[i]==0) { # two people  
 threshold[i] <- 16815  
 } else if (dat$SPORDER[i]==2 & dat$AGEP[i] < 65 & dat$OC[i]==1) {  
 threshold[i] <- 17308  
 } else if (dat$SPORDER[i]==2 & dat$AGEP[i] >= 65 & dat$OC[i]==0) {  
 threshold[i] <- 15178  
 } else if (dat$SPORDER[i]==2 & dat$AGEP[i] >= 65 & dat$OC[i]==1) {  
 threshold[i] <- 17242  
 } else if (dat$SPORDER[i]==2) { # OC is NA value  
 threshold[i] <- 16247  
 } else if (dat$SPORDER[i]==3 & dat$OC[i]==0) { # three people  
 threshold[i] <- 19642  
 } else if (dat$SPORDER[i]==3 & dat$OC[i]==1) {  
 threshold[i] <- (20212+20231)/2  
 } else if (dat$SPORDER[i]==3) { # OC is NA value  
 threshold[i] <- 19985  
 } else if (dat$SPORDER[i]==4 & dat$OC[i]==0) { # four people  
 threshold[i] <- 25900  
 } else if (dat$SPORDER[i]==4 & dat$OC[i]==1) {  
 threshold[i] <- (26324+25465+25554)/3  
 } else if (dat$SPORDER[i]==4) { # OC is NA value  
 threshold[i] <- 25701  
 } else if (dat$SPORDER[i]==5 & dat$OC[i]==0) { # five people  
 threshold[i] <- 31234  
 } else if (dat$SPORDER[i]==5 & dat$OC[i]==1) {  
 threshold[i] <- (31689+30718+29967+29509)/4  
 } else if (dat$SPORDER[i]==5) { # OC is NA value  
 threshold[i] <- 30459  
 } else if (dat$SPORDER[i]==6 & dat$OC[i]==0) { # six people  
 threshold[i] <- 35925  
 } else if (dat$SPORDER[i]==6 & dat$OC[i]==1) {  
 threshold[i] <- (36068+35324+34612+33553+32925)/5  
 } else if (dat$SPORDER[i]==6) { # OC is NA value  
 threshold[i] <- 34533  
 } else if (dat$SPORDER[i]==7 & dat$OC[i]==0) { # seven people  
 threshold[i] <- 41336  
 } else if (dat$SPORDER[i]==7 & dat$OC[i]==1) {  
 threshold[i] <- (4159+40705+40085+38929+37581+36102)/6  
 } else if (dat$SPORDER[i]==7) { # OC is NA value  
 threshold[i] <- 39194  
 } else if (dat$SPORDER[i]==8 & dat$OC[i]==0) { # eight people  
 threshold[i] <- 46231  
 } else if (dat$SPORDER[i]==8 & dat$OC[i]==1) {  
 threshold[i] <- (46640+45800+45064+44021+42696+41317+40967)/7  
 } else if (dat$SPORDER[i]==8) { # OC is NA value  
 threshold[i] <- 43602  
 } else if (dat$SPORDER[i]>=9 & dat$OC[i]==0) { # nine or more people  
 threshold[i] <- 55613  
 } else if (dat$SPORDER[i]>=9 & dat$OC[i]==1) {  
 threshold[i] <- (55883+55140+54516+53491+52082+50807+50491+48546)/8  
 } else if (dat$SPORDER[i]>=9) { # OC is NA value  
 threshold[i] <- 51393  
 } else {  
 threshold[i] <- NA  
 }  
 } # individually assign poverty threshold  
 return(threshold)  
}  
PovertyThreshold <- getThreshold(PovertyThreshold)  
dat$IncomePovertyRatio <- (dat$PERNP + dat$PINCP)/PovertyThreshold  
  
dat$IncomePovertyRatio <- dat$IncomePovertyRatio + 1 + abs(min(dat$IncomePovertyRatio, na.rm=na.omit)) # ensure all values are positive

# function for catching error in rlasso()  
myTryCatch <- function(expr) {  
 warn <- err <- NULL  
 value <- withCallingHandlers(  
 tryCatch(expr, error=function(e) {  
 err <<- e  
 NULL  
 }), warning=function(w) {  
 warn <<- w  
 invokeRestart("muffleWarning")  
 })  
 list(error=err)  
}