## Regression Without Potential Endogeneity

## Model 2 Post-Double ML OLs with exogenous variable selection ---------------------  
# manually delete potentially endogenous variables  
# Can X affect or cause Income or Poverty of Both?  
endog <- c("SPORDER", # household size  
 "CIT3","CIT4","CIT5", # citizenship status  
 "COW2","COW3","COW4","COW5","COW6","COW7","COW8", # class of worker  
 "DDRS2","DEYE2","DPHY2","DREM2", # disability  
 "ENG2","ENG3","ENG4","FER1","FER2", # level of english and child birth  
 "GCL2","GCR2", # grandparents with grandchildren  
 "HINS12","HINS22","HINS42","HINS52","HINS62","HINS72", # insurance  
 "MAR2","MAR3","MAR4","MARHD2","MARHT2","MARHT3", # marriage  
 "MIG2","MIG3", # migration  
 "NWAB2","NWAV5","NWLA3","NWLK3","NWRE2", # current work status  
 "RELP01","RELP02","RELP03","RELP04","RELP05","RELP06","RELP07", # relationship in household  
 "RELP08","RELP09","RELP10","RELP11","RELP12","RELP13","RELP15","RELP17"  
 )  
endog2 <- c(61:79,80,81:86, # degree, sex, work  
 91,92:96,97, # disability, num cars per ppl, marriage status,   
 102:106,107:108, # race, stem degree  
 116:122,123:124,125, # stem\*degree, age\*stuff, disability  
 126:134, # marriage, work, race, age\*stuff, citizenship, disability, work  
 136:140, # school, veteran, grandparents with grandchild  
 142:144, # insurance, work, school  
 148:152 # age\*stuff, grandparents with grandchild, work  
 )  
  
union <- union[-endog2]  
union <- union[-which(union %in% endog)] # delete them from formula  
  
# rewrite formula for OLS  
exogunionf <- paste(union, collapse = "+")  
exogformula <- paste(c("log(IncomePovertyRatio)", exogunionf), collapse = "~")  
  
# Training OLS regression post LASSO  
olsDLasso2 <- lm(exogformula, data = dattemp[train,])  
DMLresult2 <- summary(olsDLasso2)  
# Post-Double LASSO OLS only on Exogeneous vars Result  
DMLresult2

##   
## Call:  
## lm(formula = exogformula, data = dattemp[train, ])  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.34876 -0.39110 -0.04425 0.33410 3.11596   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.428e+00 1.631e-01 -8.758 < 2e-16 \*\*\*  
## PWGTP -1.671e-04 8.701e-06 -19.200 < 2e-16 \*\*\*  
## AGEP 8.362e-03 9.287e-05 90.045 < 2e-16 \*\*\*  
## LANX2TRUE 5.025e-02 2.234e-03 22.496 < 2e-16 \*\*\*  
## MARHYP 1.625e-03 7.909e-05 20.547 < 2e-16 \*\*\*  
## DECADE3TRUE 1.416e-01 7.867e-03 18.003 < 2e-16 \*\*\*  
## DECADE4TRUE 1.505e-01 5.346e-03 28.160 < 2e-16 \*\*\*  
## DECADE7TRUE 2.341e-02 3.879e-03 6.034 1.6e-09 \*\*\*  
## DECADE8TRUE -5.670e-02 4.600e-03 -12.326 < 2e-16 \*\*\*  
## PAOC1TRUE -1.810e-01 3.869e-03 -46.789 < 2e-16 \*\*\*  
## PAOC2TRUE -2.695e-01 2.217e-03 -121.533 < 2e-16 \*\*\*  
## PAOC4TRUE -3.561e-01 1.553e-03 -229.339 < 2e-16 \*\*\*  
## QTRBIR3TRUE 4.710e-03 1.610e-03 2.926 0.00344 \*\*   
## WAOB2TRUE -1.658e-01 9.589e-03 -17.294 < 2e-16 \*\*\*  
## WAOB3TRUE -3.119e-01 3.194e-03 -97.644 < 2e-16 \*\*\*  
## WAOB5TRUE 8.521e-02 4.784e-03 17.810 < 2e-16 \*\*\*  
## WAOB6TRUE -1.018e-01 7.735e-03 -13.166 < 2e-16 \*\*\*  
## WAOB7TRUE 2.027e-01 1.084e-02 18.696 < 2e-16 \*\*\*  
## WAOB8TRUE 4.730e-02 1.933e-02 2.448 0.01439 \*   
## AGEP\_HINS31 -2.175e-04 3.219e-04 -0.676 0.49922   
## DECADE5TRUE 9.779e-02 4.130e-03 23.680 < 2e-16 \*\*\*  
## HINS32TRUE 1.916e-01 2.207e-02 8.682 < 2e-16 \*\*\*  
## DECADE1TRUE 7.548e-02 4.239e-02 1.781 0.07499 .   
## DECADE2TRUE 7.733e-02 1.371e-02 5.640 1.7e-08 \*\*\*  
## PAOC3TRUE -2.705e-01 4.101e-03 -65.946 < 2e-16 \*\*\*  
## QTRBIR2TRUE 5.347e-03 1.649e-03 3.241 0.00119 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.5889 on 781343 degrees of freedom  
## Multiple R-squared: 0.106, Adjusted R-squared: 0.106   
## F-statistic: 3708 on 25 and 781343 DF, p-value: < 2.2e-16

# Test Prediction  
pred.olsDLasso.2 <- predict(olsDLasso2, newdata = dattemp[-train,])  
summary(pred.olsDLasso.2)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.402 2.113 2.234 2.258 2.431 2.950

length(na.omit(pred.olsDLasso.2)) # count remaining observations

## [1] 195343

# test error  
mse.2 <- mean((pred.olsDLasso.2-log(dattemp[-train,]$IncomePovertyRatio))^2, na.rm=T)  
mse.2

## [1] 0.34607

## Result 2: Analysis & Hypothesis Testing

# 3 Ways of getting Test R2  
y2 <- log(dattemp[-train,]$IncomePovertyRatio)-mean(log(dattemp[-train,]$IncomePovertyRatio))  
yhat2 <- pred.olsDLasso.2-mean(pred.olsDLasso.2)  
u2 <- y2 - yhat2  
# 1:  
# R2 = yhat\*y/yTy  
r2\_1\_2 <- (yhat2 %\*% yhat2)/(y2 %\*% y2)  
r2\_1\_2

## [,1]  
## [1,] 0.1064778

# 2:  
# R2 = 1- SSR/SST = 1- uTu/yTy  
r2\_2\_2 <- 1 - (u2 %\*% u2)/(y2 %\*% y2)  
r2\_2\_2

## [,1]  
## [1,] 0.1028011

# 3:  
# R2 = corr(y, yhat)^2, "fair r-squared"  
r2\_3\_2 <- cor.test(y2, yhat2, use = "complete.obs")  
# now, square the correlation coefficient  
r2\_3\_2

##   
## Pearson's product-moment correlation  
##   
## data: y2 and yhat2  
## t = 149.63, df = 195341, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.3166914 0.3246485  
## sample estimates:  
## cor   
## 0.3206756

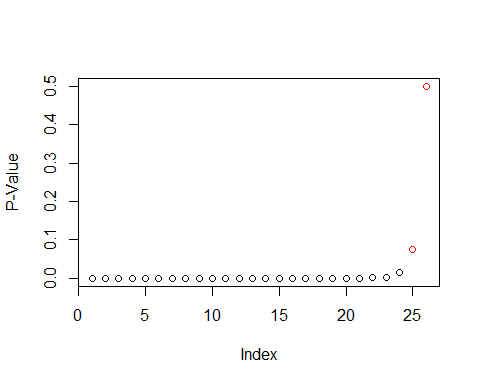
r2\_3\_2$estimate^2

## cor   
## 0.1028329

# BP test for heteroskedasticity  
bpres2 <- bptest(olsDLasso2, data = dattemp[-train,]) #reject homoskedasticity if p-value is small  
bpres2

##   
## studentized Breusch-Pagan test  
##   
## data: olsDLasso2  
## BP = 13044, df = 25, p-value < 2.2e-16

# False Discovery Rate control  
p2 <- as.data.frame(DMLresult2$coefficients[,4])  
sigcode2 <- cut(p2[,1], breaks = c(-Inf, 0.001, 0.01, 0.05, 0.1, 1),   
 labels = c("\*\*\*", "\*\*", "\*", ".", " "))  
p2$"" <- sigcode2  
  
# sort by increasing p-value  
p2 <- p2[order(p2$`DMLresult2$coefficients[, 4]`),]  
p2$BY <- 0  
m2 <- nrow(p2)  
Q = 0.10 # 10%  
cm=0  
for (ii in 1:m2) {  
 cm = cm + 1/ii  
 p2[ii,3] <- ii/m2/cm\*Q  
}  
noreject2 <- (!(p2[,1] < p2[,3]))  
plot(p2$`DMLresult2$coefficients[, 4]`,ylab="P-Value", col = ifelse(noreject2,'red','black'))



noreject2 <- which(noreject2)  
p2 <- p2[noreject2,] # these one's we cannot reject the null  
names(p2) <- c("p-value","Sig. Level","BY Stat")  
p2

## p-value Sig. Level BY Stat  
## DECADE1TRUE 0.07498765 . 0.02519782  
## AGEP\_HINS31 0.49921838 0.02594424

# get BY-adjusted p-values  
pBY2 <- as.data.frame(p.adjust(p2[,1], method = "BY")) #Benjamini-Yekutieli  
rownames(pBY2) <- rownames(p2)  
adjsigcode <- cut(pBY2[,1], breaks = c(-Inf, 0.001, 0.01, 0.05, 0.1, 1),   
 labels = c("\*\*\*", "\*\*", "\*", ".", " "))  
pBY2$"" <- adjsigcode  
  
# compare p-values for non-rejected  
fdr2 <- cbind.data.frame(p2[,c(1,2)], pBY2)  
colnames(fdr2) <- c("Original","Sig. Level", "FDR Adj.","Sig. Level")  
fdr2

## Original Sig. Level FDR Adj. Sig. Level  
## DECADE1TRUE 0.07498765 . 0.2249630   
## AGEP\_HINS31 0.49921838 0.7488276