```r
rm(list = ls());

require(devtools);
require(DataAnalytics);
require(plyr);
require(ggplot2);
require(Matrix);
require(AER);
require(sandwich);

set.seed(11);
#allows for reproducability

#adapted from Rossi tsls funciton
#In: focal variables mX, exogeneous covariates mW, intstrumental variables mZ, y variables
#out: coeficients of the regression and a matrix of modified white standard errors
twoSLS = function(mX, mW, mZ, vY) {
    #assumes both Z and W are exogeneous
    mZa = cbind(mZ, mW);
    mRInv = backsolve(chol(crossprod(mZa)), diag(ncol(mZa)));
    mPZa = tcrossprod(mZa %*% mRInv);
    mXTild = cbind(mPZa %*% mX, mW);
    mXspXsinv = chol2inv(chol(crossprod(mXTild)));

    mXW = cbind(mX, mW);
    vB = mXspXsinv %*% crossprod(mXTild, vY);

    rownames(vB) = c(colnames(mX), colnames(mW));
    resid = vY - mXW %*% vB;
    mWhiteErrors = GetMWhiteErrors(mXTild, resid);
    colnames(mWhiteErrors)= rownames(vB)

    return(list(
    vCoef = vB,
    mMWSE = mWhiteErrors,
    mHSSE = rep(var(resid),3) * mXspXsinv));
}

#in: X variables and residuals
#out: Modified White stnadard errors
GetMWhiteErrors = function(X, vResid) {
    #extract the temporary variables
    qrX = qr(X);
    mX = qr.X(qrX);
    mR = qr.R(qrX);
    mQ = qr.Q(qrX);

    mQRTInv = mQ %*% t(backsolve(mR,diag(ncol(mR))));
    return(t(mQRTInv * rep((vResid / (1 - rowSums(mQ * mQ))) ^ 2, ncol(mX))) %*% mQRTInv);
```

```r
}

#in: a regression object
#out: a matrix object for the coefficients
GetCoefAsMatrix = function(reg) matrix(coef(reg),
    nrow = length(coef(reg)),
    dimnames = list(names(coef(reg))));


tbAcemoglu = read.table("acemoglu.dat", header = TRUE);
iNumPts = nrow(tbAcemoglu);

mX = matrix(tbAcemoglu[["Exprop"]], nrow = iNumPts, dimnames = dimnames
  (tbAcemoglu["Exprop"]));
mW = cbind(matrix(tbAcemoglu[["Latitude"]], nrow = iNumPts, dimnames = dimnames
  (tbAcemoglu["Latitude"])),
    matrix(1, nrow = iNumPts));
mZ = matrix(log(tbAcemoglu[["Mort"]]), nrow = iNumPts, dimnames = dimnames
  (tbAcemoglu["Mort"]));
vY = tbAcemoglu$GDP;
colnames(mW)[ncol(mW)] = "Intercept";

spec = as.formula("GDP ~ Exprop + Latitude");
vOLS = lm(spec, data = tbAcemoglu);
vOLSCoef = GetCoefAsMatrix(vOLS);
vOLSResid = vY - cbind(mX, mW) %*% rbind(vOLSCoef[3],vOLSCoef[1],vOLSCoef[2]);
print("OLS Coeficients");
print(t(vOLSCoef));

print("OLS Homoskedastic SE");
print(sqrt(diag(vcov(vOLS))));

print("OLS Modified White Standard Errors")
mOLSMWSE = sqrt(diag(GetMWhiteErrors(cbind(mX, mW), vOLSResid)));
print(cbind(mOLSMWSE[3], mOLSMWSE[1], mOLSMWSE[2]));

#Prints in order of X, W, intercept
lSLSOut = twoSLS(mX = mX, mW = mW, mZ = mZ, vY = vY);
##Coeficients:
print("2SLS Coeficients")
print(t(lSLSOut$vCoef));

print("2SLS Homoskedastic Standard Error");
print(sqrt(diag(lSLSOut$mHSSE)));

print("2SLS Modified White Standard Error");
print(colnames(lSLSOut$mMWSE));
print(sqrt(diag(lSLSOut$mMWSE)));

#check
#aerReg = ivreg(formula = GDP ~ Exprop + Latitude | Latitude + log(Mort), data =
  tbAcemoglu);
```

```
#print(summary(aerReg));
#print(sqrt(diag(vcovHC(aerReg))));
```