Exercises 12.22

Import data

First, Set the environment and import data. *Note that I wrote the code using R*

```
rm(list=ls())
library(dplyr) #dataframe management
library(haven) #library for importing data set
library(psych) #library for trace matrix
ajr <- read_dta("AJR2001.dta")
```

(a) Estimate the OLS regression (12.86), the reduced form regression (12.87), and the 2SLS regression (12.88). (Which point estimate is different by 0.01 from the reported values? This is a common phenomenon in empirical replication).

- OLS regression.

```
Y <- as.matrix(ajr$loggdp)
X <- as.matrix(ajr %>%
                 select(risk) %>%
                 bind_cols(data.frame(con=rep(1,nrow(ajr)))))
beta_hat <- solve((t(X) %*% X)) %*% (t(X) %*% Y)
e_hat <- Y - X %*% beta_hat
row.names(beta_hat) <- c('risk','intercept')
colnames(beta_hat) <- c('coefficient')
colnames(e_hat) <- c('error')
```

Set the variable for convenience.

```
XX <- solve((t(X) %*% X)) #XX is variable name it means inverse matrix of (X'X)
leverage <- diag(X%*%XX%*%t(X))
```

Calculate S.E. using the HC3 method.

```
u3 <- X * ((e_hat/(1-leverage))%*% matrix(1,1,ncol(X)))
v3 <- XX %*% (t(u3) %*% u3) %*% XX
s3 <- sqrt(diag(v3))
```

Table 1: Coefficient estimates and Robust S.E.

	Coefficient	Robust standard errors
risk	0.52	0.05
intercept	4.69	0.33

Thus estimated OLS estimates (intercept omitted) are

$$\widehat{log(GDP\ per\ Capita)} = \underset{(0.05)}{0.52}\ risk. \tag{1}$$

- Reduced form regression.

```
Yr <- as.matrix(ajr$risk)
Xr <- as.matrix(ajr %>%
                  select(logmort0) %>%
                  bind_cols(data.frame(con=rep(1,nrow(ajr)))))
beta_hatr <- solve((t(Xr) %*% Xr)) %*% (t(Xr) %*% Yr)
e_hatr <- Yr - Xr %*% beta_hatr
row.names(beta_hatr) <- c('mortality','intercept')
colnames(beta_hatr) <- c('coefficient')
colnames(e_hatr) <- c('error')
```

Set the variable for convenience.

```
XXr <- solve((t(Xr) %*% Xr)) #XX is variable name it means inverse matrix of (X'X)
leverager <- diag(Xr%*%XXr%*%t(Xr))
```

Calculate S.E. using the HC3 method.

```
u3r <- Xr * ((e_hatr/(1-leverager))%*% matrix(1,1,ncol(Xr)))
v3r <- XXr %*% (t(u3r) %*% u3r) %*% XXr
s3r <- sqrt(diag(v3r))
```

Table 2: Coefficient estimates and Robust S.E.

	Coefficient	Robust standard errors
$log(mortality)$	-0.61	0.16
intercept	9.37	0.75

Thus first-stage regression (intercept omitted) is

$$risk = \underset{(0.16)}{-0.61}\ log(mortality) + \hat{u}. \tag{2}$$

- 2SLS regression.

```
beta_hat2sls <- solve(t(X)%*%Xr%*%solve(t(Xr)%*%Xr)%*%t(Xr)%*%X)%*%t(X)%*%Xr%*%
  solve(t(Xr)%*%Xr)%*%t(Xr)%*%Y
e_hat2sls <- Y - X %*% beta_hat2sls
row.names(beta_hat2sls) <- c('risk','intercept')
colnames(beta_hat2sls) <- c('coefficient')
colnames(e_hat2sls) <- c('error')
```

Calculate S.E. assume heteroskedastic.

```
u2sls <- Xr*(e_hat2sls%*%matrix(1,1,ncol(Xr)))
qzx <- t(Xr)%*%X
qxx <- t(X)%*%X
qzz <- t(Xr)%*%Xr
v2sls <- solve(t(qzx)%*%solve(qzz)%*%qzx)%*%t(qzx)%*%solve(qzz)%*%
  (t(u2sls)%*%u2sls)%*%solve(qzz)%*%qzx%*%solve(t(qzx)%*%solve(qzz)%*%qzx)
se2 <- sqrt(diag(v2sls))
```

Table 3: Coefficient estimates and Robust S.E.

	Coefficient	Robust standard errors
risk	0.93	0.17
intercept	1.99	1.13

Thus 2SLS regression (intercept omitted) is

$$log(GDP\ \widehat{per}\ Capita) = \underset{(0.17)}{0.93}\ risk. \tag{3}$$

From equation (3); coefficient is 0.93, We can observe that point estimate is different by 0.01 from (12.88); coefficient is 0.94.

(b) For the above estimates calculate both homoskedastic and heteroskedastic-robust standard errors. Which were used by the authors (as reported in (12.86)-(12.87)-(12.88)?)

- Calculation of homoskedastic standard errors.

```
sig2_1 <- as.numeric((t(e_hat)%*%e_hat)/(nrow(Y)-ncol(X))) #sigma for ols
sig2_2 <- as.numeric((t(e_hatr)%*%e_hatr)/(nrow(Yr)-ncol(Xr))) #sigma for first-stage
sig2_3 <- as.numeric((t(e_hat2sls)%*%e_hat2sls)/(nrow(Y)-ncol(X))) #sigma for 2sls

v0_1 <- XX*sig2_1 #variance for ols
v0_2 <- XXr*sig2_2 #variance for first-stage
v0_3 <- solve(t(qzx)%*%solve(qzz)%*%qzx)%*%t(qzx)%*%solve(qzz)%*% #variance for 2sls
  solve(qzz)%*%qzx%*%solve(t(qzx)%*%solve(qzz)%*%qzx)*sig2_3


s0_1 <- diag(sqrt(v0_1))
s0_2 <- diag(sqrt(v0_2))
s0_3 <- diag(sqrt(v0_3))


result_d <- data.frame('homoskedastic'=c(s0_1[1],s0_2[1],s0_3[1]),
                       'heteroskedastic'=c(s3[1],s3r[1],se2[1,]),
                       'reports from paper' =c(0.06,0.13,0.16),
                       row.names = c('ols','first-stage','2sls'))
```

Table 4: Homoskedastic and heteroskedastic S.E.

	homoskedastic	heteroskedastic	reports from paper
ols	0.0625186	0.0527083	0.06
first-stage	0.1269412	0.1601957	0.13
2sls	0.0746264	0.1700872	0.16

Above table is homoskedastic error and heteroskedastic-robust standard errors which calculated in the (a).

Reported S.E. of ols is more close to the homoskedastic error, But reported S.E. of 2SLS is more close to the heteroskedastic-robust Standard errors. S.E. of first-stage is almost same between homoskedastic and heteroskedastic standard errors.

(c) Calculate the 2SLS estimates by the Indirect Least Squares formula. Are they the same?

ILS estimator is

$$\hat{\beta}_{ils} = \hat{\hat{\Gamma}}^{-1}\hat{\lambda} = ((Z'Z)^{-1}Z'X)^{-1}(Z'Z)^{-1}Z'Y_1.$$

Where Z = Exogenous variables, X = Endogenous variables. Thus Calculation of ILS estimator is following codes.

```
beta_hatils <- solve(solve(t(Xr)%*%Xr)%*%t(Xr)%*%X)%*%solve(t(Xr)%*%Xr)%*%t(Xr)%*%Y
row.names(beta_hatils) <- c('risk','intercept')
colnames(beta_hatils) <- c('coefficient')
```

I will compare the result in the (e).

(d) Calculate the 2SLS estimates by the two-stage approach. Are they the same?

2SLS is calculated from (a), Thus codes are same too.

```
beta_hat2sls <- solve(t(X)%*%Xr%*%solve(t(Xr)%*%Xr)%*%t(Xr)%*%X)%*%t(X)%*%Xr%*%
  solve(t(Xr)%*%Xr)%*%t(Xr)%*%Y
row.names(beta_hat2sls) <- c('risk','intercept')
colnames(beta_hat2sls) <- c('coefficient')
```

I will compare the result in the (e).

(e) Calculate the 2SLS estimates by the control variable approach. Are they the same?

Estimator from control variable approach is

$$\hat{\beta} = (X'P_ZX)^{-1}(X'P_ZY).$$

Where $P_Z = Z(Z'Z)^{-1}Z'$.

```
Pz <- Xr%*%solve(t(Xr)%*%Xr)%*%t(Xr)
beta_hatcf <- solve(t(X)%*%Pz%*%X)%*%(t(X)%*%Pz%*%Y)
row.names(beta_hatcf) <- c('risk','intercept')
colnames(beta_hatcf) <- c('coefficient')
result_e <- bind_cols(beta_hatils,beta_hat2sls,beta_hatcf)
```

```
colnames(result_e) <- c('ILS','2SLS','Control Variable')
```

Table 5: Coefficient estimates calculated by various method

	ILS	2SLS	Control Variable
risk	0.9294897	0.9294897	0.9294897
intercept	1.9942956	1.9942956	1.9942956

As the table 5, Estimated results from ILS, 2SLS, and Control variable approach are same.

(f) Acemoglu, Johnson, and Robinson (2001) reported many specifications including alternative regressor controls, for example *latitude* and *africa*. Estimate by least squares the equation for logGDP adding *latitude* and *africa* as regressors. Does this regression suggest that latitude and africa are predictive of the level of GDP?

- Calculate OLS Regression

```
Y <- as.matrix(ajr$loggdp)
X <- as.matrix(ajr %>%
                 select(risk, latitude, africa ) %>%
                 bind_cols(data.frame(con=rep(1,nrow(ajr)))))

#calculate beta

beta_hat <- solve((t(X) %*% X)) %*% (t(X) %*% Y)
e_hat <- Y - X %*% beta_hat

#calculate robust S.E.

XX <- solve((t(X) %*% X)) #XX is variable name it means inverse matrix of (X'X)
leverage <- diag(X%*%XX%*%t(X))
u3 <- X * ((e_hat/(1-leverage))%*% matrix(1,1,ncol(X)))
v3 <- XX %*% (t(u3) %*% u3) %*% XX
s3 <- sqrt(diag(v3))
```

Table 6: Coefficient estimates and Robust S.E.

	Coefficient	Robust standard errors
risk	0.38	0.07
latitude	1.38	0.68
africa	-0.72	0.18
intercept	5.65	0.41

We can do t-test to confirm that *latitude* and *africa* are predictive of the the level of GDP. We can construct following hypothesis test. with size $\alpha = 0.05$

$$H_0 : \beta_{africa} = 0$$
$$H1 : \beta_{africa} \neq 0$$

and other test, with size $\alpha = 0.05$

$$H_0 : \beta_{latitude} = 0$$
$$H1 : \beta_{latitude} \neq 0$$

Since $t_{latitude} = 2.029412$, and $t_{africa} = 4$ we can reject null hypothesis at size of 0.05. Therefore *latitude* and *africa* are predictive of the the level of GDP.

(g) Now estimate the same equation as in (f) but by 2SLS using log(*mortality*) as an instrument for *risk*. How does the interpretation of the effect of *latitude* and *africa* change?

- 2SLS regression.

```
Xr <- as.matrix(ajr %>%
                 select(logmort0, latitude , africa) %>%
                 bind_cols(data.frame(con=rep(1,nrow(ajr)))))


beta_hat2sls <- solve(t(X)%*%Xr%*%solve(t(Xr)%*%Xr)%*%t(Xr)%*%X)%*%t(X)%*%Xr%*%
  solve(t(Xr)%*%Xr)%*%t(Xr)%*%Y
e_hat2sls <- Y - X %*% beta_hat2sls
```

Calculate S.E. assume heteroskedastic.

```
u2sls <- Xr*(e_hat2sls%*%matrix(1,1,ncol(Xr)))
qzx <- t(Xr)%*%X
qxx <- t(X)%*%X
qzz <- t(Xr)%*%Xr
v2sls <- solve(t(qzx)%*%solve(qzz)%*%qzx)%*%t(qzx)%*%solve(qzz)%*%
  (t(u2sls)%*%u2sls)%*%solve(qzz)%*%qzx%*%solve(t(qzx)%*%solve(qzz)%*%qzx)
se2 <- sqrt(diag(v2sls))
```

Table 7: Coefficient estimates and Robust S.E.

	Coefficient	Robust standard errors
risk	0.80	0.27
latitude	-0.06	1.12
africa	-0.35	0.33
intercept	3.00	1.74

No it seems not that *latitude* and *africa* are predictive of the level of GDP. As the results from table 7, the S.E for coefficients of *latitude* and *africa* are too large to argue that these variables are predictive of the level of GDP at size of 0.05.

(h) Return to our baseline model (without including *latitude* and *africa*). The authors' reduced form equation uses log(*mortality*) as the instrument, rather than, say, the level of mortality. Estimate the reduced form for risk with mortality as the instrument. (This variable is not provided in the dataset so you need to take the exponential of log(*mortality*).) Can you explain why the authors preferred the equation with log(*mortality*)?

- Reduced form regression.

```
Yr <- as.matrix(ajr$risk)
Xr <- as.matrix(ajr %>%
                  mutate(mortality = exp(logmort0)) %>%
                  select(mortality) %>%
                  bind_cols(data.frame(con=rep(1,nrow(ajr)))))
beta_hatr <- solve((t(Xr) %*% Xr)) %*% (t(Xr) %*% Yr)
e_hatr <- Yr - Xr %*% beta_hatr
row.names(beta_hatr) <- c('mortality','intercept')
colnames(beta_hatr) <- c('coefficient')
colnames(e_hatr) <- c('error')
```

Set the variable for convenience.

```
XXr <- solve((t(Xr) %*% Xr)) #XX is variable name it means inverse matrix of (X'X)
leverager <- diag(Xr%*%XXr%*%t(Xr))
```

Calculate S.E. using the HC3 method.

```
u3r <- Xr * ((e_hatr/(1-leverager))%*% matrix(1,1,ncol(Xr)))
v3r <- XXr %*% (t(u3r) %*% u3r) %*% XXr
s3r <- sqrt(diag(v3r))
```

Table 8: Coefficient estimates and Robust S.E.

	Coefficient	Robust standard errors
mortality	0.00	0.0
intercept	6.71	0.2

Thus first-stage regression (intercept omitted) is

$$risk = \underset{(0)}{0} \times mortality + \hat{u}. \tag{4}$$

As the equation 4, We cannot find correlation between risk and mortality. By taking log to mortality, We can observe correlation between risk and log(mortality). Thus I can guess that this is reason why authors preferred the equation with log(mortality).

(i) Try an alternative reduced form including both log(*mortality*) and the square of log(*mortality*). Interpret the results. Re-estimate the structural equation by 2SLS using both log(*mortality*) and its square as instruments. How do the results change?

- Reduced form regression.

```
Yr <- as.matrix(ajr$risk)
Y <- as.matrix(ajr$loggdp)
X <- as.matrix(ajr %>%
                 select(risk) %>%
                 bind_cols(data.frame(con=rep(1,nrow(ajr)))))
Xr <- as.matrix(ajr %>%
                 select(logmort0) %>%
                 mutate(logmort0sq = logmort0^2) %>%
                 bind_cols(data.frame(con=rep(1,nrow(ajr)))))
beta_hatr <- solve((t(Xr) %*% Xr)) %*% (t(Xr) %*% Yr)
```

```
e_hatr <- Yr - Xr %*% beta_hatr
```

Set the variable for convenience.

```
XXr <- solve((t(Xr) %*% Xr)) #XX is variable name it means inverse matrix of (X'X)
leverager <- diag(Xr%*%XXr%*%t(Xr))
```

Calculate S.E. using the HC3 method.

```
u3r <- Xr * ((e_hatr/(1-leverager))%*% matrix(1,1,ncol(Xr)))
v3r <- XXr %*% (t(u3r) %*% u3r) %*% XXr
s3r <- sqrt(diag(v3r))
```

Table 9: Coefficient estimates and Robust S.E.

	Coefficient	Robust standard errors
$log(mortality)$	-2.65	0.95
$log(mortality)^2$	0.21	0.11
intercept	13.95	2.01

Thus first-stage regression (intercept omitted) is

$$risk = -\underset{(0.95)}{2.65} log(mortality) + \underset{(0.11)}{0.21} log(mortality)^2 + \hat{u}. \tag{5}$$

- 2SLS regression.

```
beta_hat2sls <- solve(t(X)%*%Xr%*%solve(t(Xr)%*%Xr)%*%t(Xr)%*%X)%*%t(X)%*%Xr%*%
  solve(t(Xr)%*%Xr)%*%t(Xr)%*%Y
e_hat2sls <- Y - X %*% beta_hat2sls
```

Calculate S.E. assume heteroskedastic.

```
u2sls <- Xr*(e_hat2sls%*%matrix(1,1,ncol(Xr)))
qzx <- t(Xr)%*%X
qxx <- t(X)%*%X
qzz <- t(Xr)%*%Xr
v2sls <- solve(t(qzx)%*%solve(qzz)%*%qzx)%*%t(qzx)%*%solve(qzz)%*%
  (t(u2sls)%*%u2sls)%*%solve(qzz)%*%qzx%*%solve(t(qzx)%*%solve(qzz)%*%qzx)
se2 <- sqrt(diag(v2sls))
```

Table 10: Coefficient estimates and Robust S.E.

	Coefficient	Robust standard errors
risk	0.77	0.10
intercept	3.02	0.67

Comparing table 3 and table 10, The coefficient of the risk in this problem is smaller than coefficient from past problem.

(j) For the estimates in (i) are the instruments strong or weak using the Stock-Yogo test?

- Calculate F statistic for Stock-Yogo test.

```
sigtildesq <-t(Yr-mean(Yr)) %*% (Yr-mean(Yr))
sigbarsq <- t(e_hatr) %*% e_hatr
F <- ((64-3)/2)* (sigtildesq- sigbarsq)/sigbarsq
F
```

```
         [,1]
[1,] 18.42273
```

5% Critical values	# of endogenous regressors : 1			
H0: Instruments are weak	# of excluded instruments : 2			
	10%	15%	20%	25%
2SLS size of nominal 5% Wald test	19.93	11.59	8.75	7.25

Table 11: 5% Critical Value for Weak Instruments

We have $F = 18.42273$ which exceeds the 15% size threshold for 2SLS as shown in table 10. Thus we can interpret the conventional 2SLS confidence interval as having coverage of 85%.

(k) Calculate and interpret a test for exogeneity of the instruments.

- OlS regression from (a).

```
Y <- as.matrix(ajr$loggdp)
X <- as.matrix(ajr %>%
                 select(risk) %>%
                 bind_cols(data.frame(con=rep(1,nrow(ajr)))))
beta_hat <- solve((t(X) %*% X)) %*% (t(X) %*% Y)
e_hat <- Y - X %*% beta_hat
row.names(beta_hat) <- c('risk','intercept')
colnames(beta_hat) <- c('coefficient')
colnames(e_hat) <- c('error')
XX <- solve((t(X) %*% X)) #XX is variable name it means inverse matrix of (X'X)
leverage <- diag(X%*%XX%*%t(X))
u3 <- X * ((e_hat/(1-leverage))%*% matrix(1,1,ncol(X)))
v3 <- XX %*% (t(u3) %*% u3) %*% XX
s3 <- sqrt(diag(v3))
```

```
pz <- Xr%*%XXr%*%t(Xr)
p1 <- 0
m1 <- diag(64) - p1
x2 <- Yr

beta2hat <- beta_hat[1]
beta2tilde <- beta_hat2sls[1]


T <- t(beta2hat - beta2tilde) %*% solve(solve(t(x2)%*%(pz-p1)%*%x2)-solve(t(x2)%*%m1%*%x2)
```