Problem Set #5

Anaya Hall and Christian Miller

5/2/2018

Part 1: Theory

(Optional – skip for now!)

Part 2: Instrumental Variables

Question 1: NLS80

Revisit the model from *Problem Set #3*, now including ability.

$log(wage) = \beta_0 + exper \cdot \beta_1 + tenure \cdot \beta_2 + married \cdot \beta_3 + south \cdot \beta_4 + urban \cdot \beta_5 + black \cdot \beta_6 + educ \cdot \beta_7 + abil \cdot \gamma + \epsilon$

```
# Read in CSV as data.frame
wage_df <- readr::read_csv("nls80.csv")

# Select only the variables in our model
wage_df %<>% select(lwage, wage, exper, tenure, married, south, urban, black, educ, iq)
```

(a) Bias of coefficient on education

Derive the bias of β_7. Show which direction the bias goes in depending on whether the correlation between ability and education is positive or negative.

$abil = \delta_0 + exper \cdot \delta_1 + tenure \cdot \delta_2 + married \cdot \delta_3 + south \cdot \delta_4 + urban \cdot \delta_5 + black \cdot \delta_6 + educ \cdot \delta_7 + \eta$

$log(wage) = (\beta_0 + \gamma\delta_0) + exper \cdot (\beta_1 + \gamma\delta_1) + tenure \cdot (\beta_2 + \gamma\delta_2) + married \cdot (\beta_3 + \gamma\delta_3) + south \cdot (\beta_4 + \gamma\delta_4) + urban \cdot (\beta_5 + \gamma\delta_5) + black \cdot (\beta_6 + \gamma\delta_6) + educ \cdot (\beta_7 + \gamma\delta_7) + \gamma\eta + v$

Assume that all δ's are zero except for the one on the variable of interest (education)

$log(wage) = \beta_0 + exper \cdot \beta_1 + tenure \cdot \beta_2 + married \cdot \beta_3 + south \cdot \beta_4 + urban \cdot \beta_5 + black \cdot \beta_6 + educ \cdot (\beta_7 + \gamma\delta_7) + \gamma\eta + v$

Where

$plimb_7 = \beta_7 + \gamma\delta_7$

$plimb_7 = \beta_7 + \gamma \cdot \frac{Cov[abil, educ]}{Var[educ]}$

Truth is β_7 , bias is $\gamma \cdot \frac{Cov[abil, educ]}{Var[educ]}$

We expect the sign on γ to be positive (higher abiltiy should lead to higher wage), the covariance of ability and education to also be positive (more able people acheive higher levels of education), and, of course, the variance of education is positive. Thus, the bias will also be *positive* (biased upward! i.e. we will over attribute the effect of education on wage).

(b) Proxy for ability

Estimate the model above excluding ability, record your parameter estimates, standard errors and R^2.

- OLS function -

First, let's load our OLS function.

```
# Function to convert tibble, data.frame, or tbl_df to matrix
to_matrix <- function(the_df, vars) {
  # Create a matrix from variables in var
  new_mat <- the_df %>%
    #Select the columns given in 'vars'
    select_(.dots = vars) %>%
    # Convert to matrix
    as.matrix()
  # Return 'new_mat'
  return(new_mat)
}


b_ols <- function(y, X) {
  # Calculate beta hat
  beta_hat <- solve(t(X) %*% X) %*% t(X) %*% y
  # Return beta_hat
  return(beta_hat)
}

ols <- function(data, y_data, X_data, intercept = T, hetsked = F, H0 = 0, two_tail = T, alpha = 0.
  # Function setup ----
    # Require the 'dplyr' package
    require(dplyr)

  # Create dependent and independent variable matrices ----
    # y matrix
    y <- to_matrix (the_df = data, vars = y_data)
    # X matrix
    X <- to_matrix (the_df = data, vars = X_data)
      # If 'intercept' is TRUE, then add a column of ones
      if (intercept == T) {
      X <- cbind(1,X)
      colnames(X) <- c("intercept", X_data)
      }

  # Calculate b, y_hat, and residuals ----
    b <- solve(t(X) %*% X) %*% t(X) %*% y
    y_hat <- X %*% b
    e <- y - y_hat

    # Inverse of X'X
    XX <- t(X) %*% X
```

```r
  XX_inv <- solve(t(X) %*% X)

  if (hetsked == T) {
    # For each row, calculate x_i' x_i e_i^2; then sum
   sigma_hat <- lapply(X = 1:n, FUN = function(i) {
    # Define x_i
    x_i <- matrix(as.vector(X[i,]), nrow = 1)
    # Return x_i' x_i e_i^2
    return(t(x_i) %*% x_i * e[i]^2)
    }) %>% Reduce(f = "+", x = .) }

  if (hetsked == F) sigma_hat <- XX

# Useful -----
  n <- nrow(X) # number of observations
  k <- ncol(X) # number of independent variables
  dof <- n - k # degrees of freedom
  i <- rep(1,n) # column of ones for demeaning matrix
  A <- diag(i) - (1 / n) * i %*% t(i) # demeaning matrix
  y_star <- A %*% y # for SST
  X_star <- A %*% X # for SSM
  SST <- drop(t(y_star) %*% y_star)
  SSM <- drop(t(b) %*% t(X_star) %*% X_star %*% b)
  SSR <- drop(t(e) %*% e)

# Measures of fit and estimated variance ----
  R2uc <- drop((t(y_hat) %*% y_hat)/(t(y) %*% y)) # Uncentered R^2
  R2 <- 1 - SSR/SST # Uncentered R^2
  R2adj <- 1 - (n-1)/dof * (1 - R2) # Adjusted R^2
  AIC <- log(SSR/n) + 2*k/n # AIC
  SIC <- log(SSR/n) + k/n*log(n) # SIC
  s2 <- SSR/dof # s^2

# Measures of fit table ----
  mof_table_df <- data.frame(R2uc, R2, R2adj, SIC, AIC, SSR, s2)
  mof_table_col_names <- c("$R^2_\\text{uc}$", "$R^2$",
                           "$R^2_\\text{adj}$",
                           "SIC", "AIC", "SSR", "$s^2$")
  mof_table <-  mof_table_df %>% knitr::kable(
    row.names = F,
    col.names = mof_table_col_names,
    format.args = list(scientific = F, digits = 4),
    booktabs = T,
    escape = F
  )

# t-test----
  # Standard error
  se <- sqrt(s2 * diag(XX_inv %*% sigma_hat %*% XX_inv)) # Vector of _t_ statistics
  # Vector of _t_ statistics
```

```r
    t_stats <- (b - H0) / se
    # Calculate the p-values
    if (two_tail == T) {
    p_values <- pt(q = abs(t_stats), df = dof, lower.tail = F) * 2
    } else {
      p_values <- pt(q = abs(t_stats), df = dof, lower.tail = F)
    }
    # Do we (fail to) reject?
    reject <- ifelse(p_values < alpha, reject <- "Reject", reject <- "Fail to Reject")

    # Nice table (data.frame) of results
    ttest_df <- data.frame(
      # The rows have the coef. names
      effect = rownames(b),
      # Estimated coefficients
      coef = as.vector(b) %>% round(3),
      # Standard errors
      std_error = as.vector(se) %>% round(4),
      # t statistics
      t_stat = as.vector(t_stats) %>% round(3),
      # p-values
      p_value = as.vector(p_values) %>% round(4),
      # reject null?
      significance = as.character(reject)
      )

    ttest_table <-  ttest_df %>% knitr::kable(
      col.names = c("", "Coef.", "S.E.", "t Stat", "p-Value", "Decision"),
      booktabs = T,
      format.args = list(scientific = F),
      escape = F,
      caption = "OLS Results"
    )

  # Data frame for exporting for y, y_hat, X, and e vectors ----
    export_df <- data.frame(y, y_hat, e, X) %>% tbl_df()
    colnames(export_df) <- c("y","y_hat","e",colnames(X))

  # Return ----
    return(list(n=n, dof=dof, b=b, se=se, vars=export_df, R2uc=R2uc,R2=R2,
                R2adj=R2adj, AIC=AIC, SIC=SIC, s2=s2, SST=SST, SSR=SSR,
                mof_table=mof_table, ttest=ttest_table))
}
```

```r
model_1 <- ols(wage_df, y_data = "lwage",
               X_data = c("exper", "tenure", "married", "south", "urban", "black", "educ"))

model_1$ttest
```

Table 1: OLS Results

	Coef.	S.E.	t Stat	p-Value	Decision
intercept	5.395	0.1132	47.653	0.0000	Reject
exper	0.014	0.0032	4.409	0.0000	Reject
tenure	0.012	0.0025	4.789	0.0000	Reject
married	0.199	0.0391	5.107	0.0000	Reject
south	-0.091	0.0262	-3.463	0.0006	Reject
urban	0.184	0.0270	6.822	0.0000	Reject
black	-0.188	0.0377	-5.000	0.0000	Reject
educ	0.065	0.0063	10.468	0.0000	Reject

```r
model_1$mof
```

R^2_{uc}	R^2	R^2_{adj}	SIC	AIC	SSR	s^2
0.9971	0.2526	0.2469	-1.963	-2.005	123.8	0.1336

(c) Include IQ

(c) Estimate the model including IQ as a proxy, record your parameter estimates, standard errors and R^2.

```r
model_iq <- ols(wage_df, y_data = "lwage",
               X_data = c("exper", "tenure", "married", "south", "urban", "black", "educ", "iq"))

model_iq$ttest
```

Table 3: OLS Results

	Coef.	S.E.	t Stat	p-Value	Decision
intercept	5.176	0.1280	40.441	0.0000	Reject
exper	0.014	0.0032	4.469	0.0000	Reject
tenure	0.011	0.0024	4.671	0.0000	Reject
married	0.200	0.0388	5.148	0.0000	Reject
south	-0.080	0.0263	-3.054	0.0023	Reject
urban	0.182	0.0268	6.791	0.0000	Reject
black	-0.143	0.0395	-3.624	0.0003	Reject
educ	0.054	0.0069	7.853	0.0000	Reject
iq	0.004	0.0010	3.589	0.0004	Reject

```r
model_iq$mof
```

R^2_{uc}	R^2	R^2_{adj}	SIC	AIC	SSR	s^2
0.9972	0.2628	0.2564	-1.97	-2.016	122.1	0.1319

(d) Returns on education.

What happens to returns to schooling? Does this result confirm your suspicion of how ability and schooling are expected to be correlated?

When we include IQ, the magnitude of the parameter estimate for the returns on education decreased, which suggests that we were correct in our guess that the estimate from the first OLS regression was upwardly biased. If IQ **is** a good proxy for ability, this does confirm our suspicion that ability is correlated with education. In the first model, some of the returns on ability (IQ) were mis-attributed to education. In the second model, we correct for this, and see that the parameter estimate on ability is indeed significant. As well, we get a better fit, R^2, when including the IQ.

Question 2: Recreate results from Card

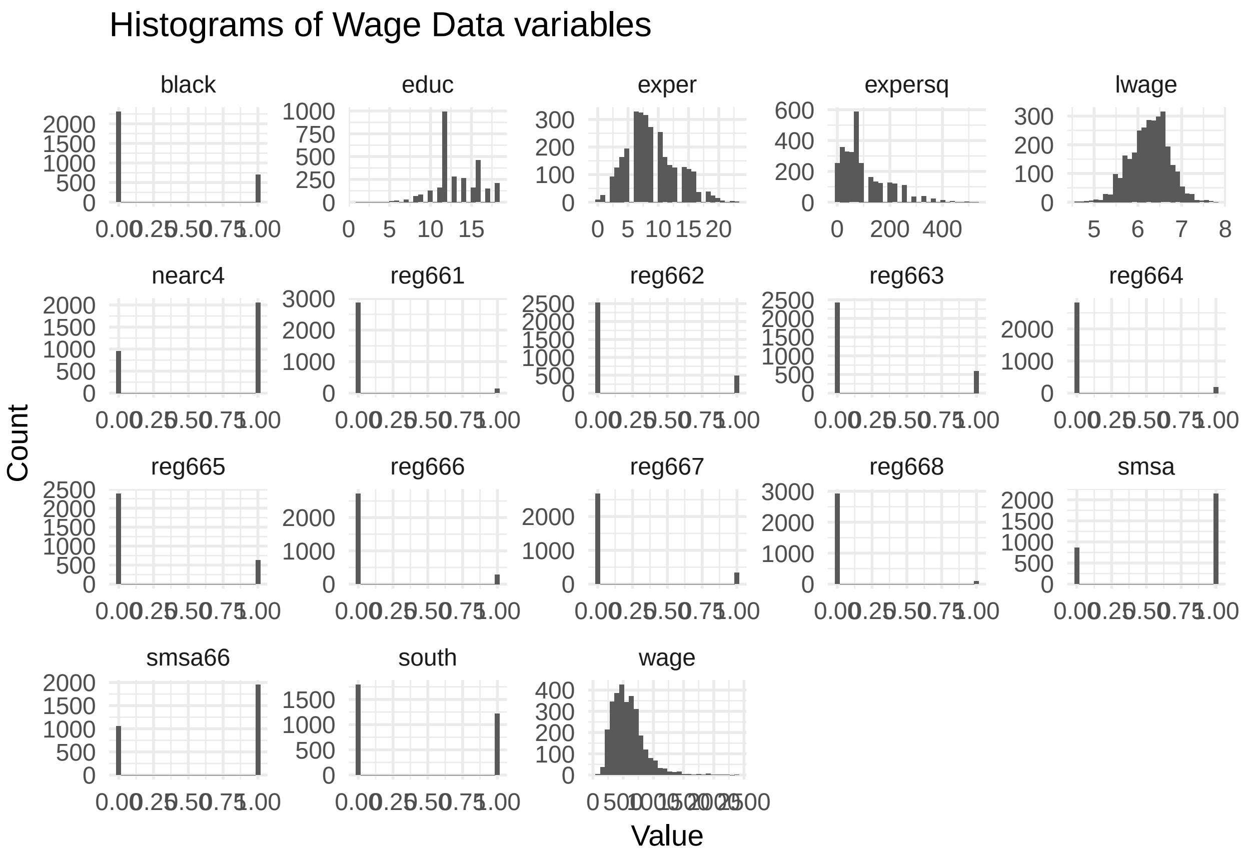
(a) Read in data & plot

```
# Read in CSV as data.frame
card_df <- readr::read_csv("card.csv")

# Select only the variables in our model
card_df %<>% select(lwage, wage, educ, exper, expersq, black, south, smsa, smsa66, reg661, reg662,

head(card_df)
```

```
## # A tibble: 6 x 19
##   lwage  wage  educ exper expersq black south  smsa smsa66 reg661 reg662
##   <dbl> <int> <int> <int>   <int> <int> <int> <int>  <int>  <int>  <int>
## 1  6.31   548     7    16     256     1     0     1      1      1      0
## 2  6.18   481    12     9      81     0     0     1      1      1      0
## 3  6.58   721    12    16     256     0     0     1      1      1      0
## 4  5.52   250    11    10     100     0     0     1      1      0      1
## 5  6.59   729    12    16     256     0     0     1      1      0      1
## 6  6.21   500    12     8      64     0     0     1      1      0      1
## # ... with 8 more variables: reg663 <int>, reg664 <int>, reg665 <int>,
## #   reg666 <int>, reg667 <int>, reg668 <int>, nearc4 <int>, nearc2 <int>
```

```
ggplot(data = gather(card_df), aes(x = value)) +
  geom_histogram() +
  facet_wrap(~ key, scales = "free") +
  ggtitle("Histograms of Wage Data variables") +
  ylab("Count") +
  xlab("Value") + theme_minimal()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



(b) OLS on log(wage)

```
rhs_vars <- c("educ", "exper", "expersq", "black", "south", "smsa", "reg661", "reg662", "reg663",

model1 <- ols(card_df, "lwage", rhs_vars)

model1$ttest
```

Table 5: OLS Results

	Coef.	S.E.	t Stat	p-Value	Decision
intercept	4.739	0.0715	66.259	0.0000	Reject
educ	0.075	0.0035	21.351	0.0000	Reject
exper	0.085	0.0066	12.806	0.0000	Reject
expersq	-0.002	0.0003	-7.223	0.0000	Reject
black	-0.199	0.0182	-10.906	0.0000	Reject
south	-0.148	0.0260	-5.695	0.0000	Reject
smsa	0.136	0.0201	6.785	0.0000	Reject
reg661	-0.119	0.0388	-3.054	0.0023	Reject
reg662	-0.022	0.0283	-0.786	0.4321	Fail to Reject
reg663	0.026	0.0274	0.949	0.3427	Fail to Reject
reg664	-0.063	0.0357	-1.780	0.0753	Fail to Reject
reg665	0.009	0.0361	0.262	0.7935	Fail to Reject
reg666	0.022	0.0401	0.547	0.5842	Fail to Reject
reg667	-0.001	0.0394	-0.015	0.9881	Fail to Reject

	Coef.	S.E.	t Stat	p-Value	Decision
reg668	-0.175	0.0463	-3.777	0.0002	Reject
smsa66	0.026	0.0194	1.349	0.1773	Fail to Reject

These point estimates are very close to those of the paper. However, we do not know how to interprent yes/no from the region (reg661-668) and live in SMSA in 1966 (smsa66) variable in comparison to our estimate values.

(c) Reduced Form

Estimate reduced form equation for *educ* containing all of the explanatory variables and the dummy variable *nearc4*

```
rhs_vars <- c("nearc4", "exper", "expersq", "black", "south", "smsa", "reg661", "reg662", "reg663'

rf <- ols(card_df, "educ", rhs_vars)

rf$ttest
```

Table 6: OLS Results

	Coef.	S.E.	t Stat	p-Value	Decision
intercept	16.849	0.2111	79.805	0.0000	Reject
nearc4	0.320	0.0879	3.641	0.0003	Reject
exper	-0.413	0.0337	-12.241	0.0000	Reject
expersq	0.001	0.0017	0.526	0.5987	Fail to Reject
black	-0.936	0.0937	-9.981	0.0000	Reject
south	-0.052	0.1354	-0.381	0.7032	Fail to Reject
smsa	0.402	0.1048	3.837	0.0001	Reject
reg661	-0.210	0.2025	-1.039	0.2991	Fail to Reject
reg662	-0.289	0.1473	-1.961	0.0500	Reject
reg663	-0.238	0.1426	-1.670	0.0950	Fail to Reject
reg664	-0.093	0.1860	-0.501	0.6167	Fail to Reject
reg665	-0.483	0.1882	-2.566	0.0103	Reject
reg666	-0.513	0.2096	-2.448	0.0144	Reject
reg667	-0.427	0.2056	-2.077	0.0379	Reject
reg668	0.314	0.2417	1.298	0.1945	Fail to Reject
smsa66	0.025	0.1058	0.241	0.8096	Fail to Reject

Yes, the partial correlation between *educ* and *nearc4* IS statistically significant!

(d) Single IV

Estimate the *log(wage)* equation by instrumental variables, using *nearc4* as an instrument for *educ*.

Compare the 95% confidence interval for the return to educutioan to that obtained from the Least Squares regression above.

```r
iv <- function(data, y_var, X_vars, Z_vars, intercept = T, hetsked = T, alpha = 0.05) {
    y <- to_matrix (the_df = data, vars = y_vars)
    X <- to_matrix (the_df = data, vars = X_vars)
    Z <- to_matrix (the_df = data, vars = Z_vars)

  # Add intercept
  if (intercept == T) X <- cbind(1, X)
  if (intercept == T) Z <- cbind(1, Z)
  # Calculate n and k for degrees of freedom
  n <- nrow(X)
  k <- ncol(X)
  # Estimate coefficients
  b <- solve(t(Z) %*% X) %*% t(Z) %*% y
  # Update names
  if (intercept == T) rownames(b)[1] <- "Intercept" # Calculate OLS residuals
  e <- y - X %*% b
  s2 <- (t(e) %*% e) / (n-k)

  # Calculate X_hat
  X_hat <- Z %*% solve(t(Z) %*% Z) %*% t(Z) %*% X
  # Calculate the inverse of X_hat'X_hat
  XX <- t(X_hat) %*% X_hat
  # Inverse of X'X
  XX_inv <- solve(XX)
  # Calculate the variance-covariance matrix
  if (hetsked == T) {
    sigma_hat <- lapply(X = 1:n, FUN = function(i) {
      # Define x_i
      x_i <- matrix(as.vector(X_hat[i,]), nrow = 1) # Return x_i' x_i e_i^2
      return(t(x_i) %*% x_i * e[i]^2)
    }) %>% Reduce(f = "+", x = .)
  }

  if (hetsked == F) sigma_hat <- XX
  # Calculate the standard error
  se <- sqrt(s2 * diag(XX_inv %*% sigma_hat %*% XX_inv)) # Vector of _t_ statistics
  t_stats <- (b - 0) / se
  # Calculate the p-values
  p_values = pt(q = abs(t_stats), df = n-k, lower.tail = F) * 2 # Names for coefficients
  var_names <- X_vars
  if (intercept == T) var_names <- c("Intercept", var_names)

    # t-test----
    # Do we (fail to) reject?
    reject <- ifelse(p_values < alpha, reject <- "Reject", reject <- "Fail to Reject")
    # Nice table (data.frame) of results
    results <- data.frame(
      # The rows have the coef. names
      effect = rownames(b),
      # Estimated coefficients
```

```r
      coef = as.vector(b) %>% round(3),
      # Standard errors
      std_error = as.vector(se) %>% round(4),
      # t statistics
      t_stat = as.vector(t_stats) %>% round(3),
      # p-values
      p_value = as.vector(p_values) %>% round(4),
      # reject null?
      significance = as.character(reject)
      )

    ttest_table <-  results %>% knitr::kable(
      col.names = c("", "Coef.", "S.E.", "t Stat", "p-Value", "Decision"),
      booktabs = T,
      format.args = list(scientific = F),
      escape = F,
      caption = "IV-OLS Results")

  return(ttest_table)
}

Z_vars <- c("nearc4", "exper", "expersq", "black", "south", "smsa", "reg661", "reg662", "reg663",
y_vars <- c("lwage")
X_vars <- c("educ", "exper", "expersq", "black", "south", "smsa", "reg661", "reg662", "reg663", "r
# # Run OLS
(iv1 <- iv(card_df, y_vars, X_vars, Z_vars, T, T))
```

```
## Warning in s2 * diag(XX_inv %*% sigma_hat %*% XX_inv): Recycling array of length 1 in array-vec
##   Use c() or as.vector() instead.
```

Table 7: IV-OLS Results

	Coef.	S.E.	t Stat	p-Value	Decision
Intercept	3.774	0.3563	10.593	0.0000	Reject
educ	0.132	0.0210	6.271	0.0000	Reject
exper	0.108	0.0091	11.942	0.0000	Reject
expersq	-0.002	0.0001	-17.286	0.0000	Reject
black	-0.147	0.0203	-7.218	0.0000	Reject
south	-0.145	0.0113	-12.818	0.0000	Reject
smsa	0.112	0.0121	9.269	0.0000	Reject
reg661	-0.108	0.0159	-6.777	0.0000	Reject
reg662	-0.007	0.0131	-0.538	0.5903	Fail to Reject
reg663	0.040	0.0126	3.203	0.0014	Reject
reg664	-0.058	0.0152	-3.804	0.0001	Reject
reg665	0.038	0.0192	2.002	0.0454	Reject
reg666	0.055	0.0202	2.721	0.0065	Reject
reg667	0.027	0.0195	1.375	0.1692	Fail to Reject
reg668	-0.191	0.0197	-9.698	0.0000	Reject
smsa66	0.019	0.0080	2.327	0.0201	Reject

Table 8: Confidence Interval- Return using nearcr as instrument

x	x
0.0903438	0.1726638

Table 9: Confidence Interval- Return on education

x	x
0.06814	0.08186

```
# Compare 95% confidence interval for return on education using nearc4 has IV to that of the OLS 

iv_b <- 0.1315038
iv_se <- 0.0210

ols_b <- 0.075
ols_se <- 0.0035

CI <- function(b, se, alpha=1.96) {
  CI <- list( (b - alpha*se), (b + alpha*se))
  return(CI)
}
```

```
CI(iv_b, iv_se) %>% knitr::kable(caption = "Confidence Interval- Return using nearcr as instrument
```

```
CI(ols_b, ols_se) %>% knitr::kable(caption = "Confidence Interval- Return on education")
```

Wider confidence intervals using *near4c* as IV than in the original model. The 95% confidence interval using the instrument is [0.0903, 0.1727], while from OLS it was [0.0681,0.0819].

(e) Multiple IV

Use *nearc2* and *nearc4* as instruments for *educ.*

First, lets build a function for two stage least squares (2SLS or TSLS) - Multiple Instruments

```
b_2sls <- function(data, y_var, X_vars, Z_vars, intercept = T) {
  # Turn data into matrices
  y <- to_matrix(data, y_var)
  X <- to_matrix(data, X_vars)
  Z <- to_matrix(data, Z_vars)
  # Add intercept
  if (intercept == T) X <- cbind(1, X)
  if (intercept == T) Z <- cbind(1, Z)
  # Estimate the first stage
  b_stage1 <- solve(t(Z) %*% Z) %*% t(Z) %*% X
  # Fit the first stage values
  X_hat <- Z %*% b_stage1
  # Estimate the second stage
  b_stage2 <- solve(t(X_hat) %*% X_hat) %*% t(X_hat) %*% y
```

```r
  # Update names
  if (intercept == T) rownames(b_stage2)[1] <- "Intercept"
  # Return beta_hat
  return(b_stage2)
}
```

```r
tsls <- function(data, y_vars, X_vars, Z_vars, intercept = T, hetsked = F) {

  # Turn data into matrices
  y <- to_matrix(data, y_vars)
  X <- to_matrix(data, X_vars)
  Z <- to_matrix(data, Z_vars)
  # Calculate n and k for degrees of freedom
  n <- nrow(X)
  k <- ncol(X)
  # Add intercept
  if (intercept == T) X <- cbind(1, X)
  if (intercept == T) Z <- cbind(1, Z)

  redform <- ols(data, y_vars, Z_vars, intercept, hetsked)$ttest

  # First stage
  b_stage1 <- solve(t(Z) %*% Z) %*% t(Z) %*% X
  # Fit the first stage values
  X_hat <- Z %*% b_stage1
  # Estimate the second stage
  b_stage2 <- solve(t(X_hat) %*% X_hat) %*% t(X_hat) %*% y

   # INCORRECT STANDARD ERRORS -- use X_hat
  e_inc <- y - X_hat %*% b_stage2
  s2_inc <- (t(e_inc) %*% e_inc) / (n-k)
  s2_inc %<>% as.numeric()
  XX_inv <- solve(t(X_hat) %*% X_hat)
  se_inc <- sqrt(s2_inc * diag(XX_inv))

  # Update names
  if (intercept == T) rownames(b_stage2)[1] <- "Intercept"


  # Calculate P_Z
  P_Z <- Z %*% solve(t(Z) %*% Z) %*% t(Z)
  # Calculate b_2sls
  b <- solve(t(X) %*% P_Z %*% X) %*% t(X) %*% P_Z %*% y
  # Calculate OLS residuals
  e <- y - X %*% b
  # Calculate s^2
  s2 <- (t(e) %*% e) / (n - k)
  s2 %<>% as.numeric()
  # Inverse of X' Pz X
  XX_inv <- solve(t(X) %*% P_Z %*% X)
```

```r
  # Standard error
  se <- sqrt(s2 * diag(XX_inv))    # These should be the 'correct' standard errors
  # Vector of _t_ statistics
  t_stats <- (b - 0) / se
  t_stats_inc <- (b - 0) / se_inc
  # Calculate the p-values
  p_values = pt(q = abs(t_stats), df = n-k, lower.tail = F) * 2
  p_values_inc = pt(q = abs(t_stats_inc), df = n-k, lower.tail = F) * 2

  # Update names
  if (intercept == T) rownames(b)[1] <- "Intercept"

  # Nice table (data.frame) of CORRECT results
  correct_res <- data.frame(
    # The rows have the coef. names
    effect = rownames(b),
    # Estimated coefficients
    coef = as.vector(b),
    # Standard errors
    std_error = as.vector(se),
    # t statistics
    t_stat = as.vector(t_stats),
    # p-values
    p_value = as.vector(p_values)
    )
  # INCORRECT RESULTS
    incorrect_res <- data.frame(
    effect = rownames(b),
    coef = as.vector(b),
    std_error = as.vector(se_inc),
    # t statistics
    t_stat = as.vector(t_stats_inc),
    # p-values
    p_value = as.vector(p_values_inc)
  )

  results_list <- list()

  # Return the results
  return(list(correctSE = correct_res, incorrectSE = incorrect_res, redform = redform))

}

Z_vars <- c("nearc4", "nearc2", "exper", "expersq", "black", "south", "smsa", "reg661", "reg662", 
y_vars <- c("lwage")
X_vars <- c("educ", "exper", "expersq", "black", "south", "smsa", "reg661", "reg662", "reg663", "r
```

```r
#RUN FUNCTION
two_stage <- tsls(data = card_df, y_vars, X_vars, Z_vars, T, F)
```

```r
# Reduced Form
two_stage$redform
```

Table 10: OLS Results

	Coef.	S.E.	t Stat	p-Value	Decision
intercept	5.968	0.0445	134.123	0.0000	Reject
nearc4	0.042	0.0181	2.340	0.0194	Reject
nearc2	0.036	0.0159	2.251	0.0245	Reject
exper	0.054	0.0069	7.807	0.0000	Reject
expersq	-0.002	0.0003	-6.562	0.0000	Reject
black	-0.273	0.0193	-14.116	0.0000	Reject
south	-0.149	0.0279	-5.332	0.0000	Reject
smsa	0.164	0.0216	7.632	0.0000	Reject
reg661	-0.123	0.0420	-2.940	0.0033	Reject
reg662	-0.039	0.0304	-1.291	0.1968	Fail to Reject
reg663	0.023	0.0300	0.771	0.4409	Fail to Reject
reg664	-0.054	0.0389	-1.389	0.1651	Fail to Reject
reg665	-0.012	0.0391	-0.299	0.7648	Fail to Reject
reg666	-0.009	0.0431	-0.214	0.8307	Fail to Reject
reg667	-0.015	0.0428	-0.350	0.7264	Fail to Reject
reg668	-0.130	0.0505	-2.570	0.0102	Reject
smsa66	0.014	0.0220	0.658	0.5103	Fail to Reject

Comment on the significance of the partial correlations of both instruments in the reduced form.

Both instruments (*nearc4* and *nearc2*) show positive and significant effects.

Show your standard errors from the second stage and compare them to the correct standard errors.

```r
two_stage$correctSE %>% knitr::kable(caption = "Correct Standard Errors")
```

Table 11: Correct Standard Errors

effect	coef	std_error	t_stat	p_value
Intercept	3.3396875	0.8943883	3.7340464	0.0001919
educ	0.1570593	0.0525695	2.9876535	0.0028341
exper	0.1188149	0.0228023	5.2106618	0.0000002
expersq	-0.0023565	0.0003475	-6.7820393	0.0000000
black	-0.1232778	0.0521413	-2.3643020	0.0181275
south	-0.1431945	0.0284400	-5.0349610	0.0000005
smsa	0.1007530	0.0315141	3.1970804	0.0014027
reg661	-0.1029760	0.0434151	-2.3718928	0.0177601
reg662	-0.0002287	0.0337886	-0.0067676	0.9946007
reg663	0.0469556	0.0326436	1.4384337	0.1504155
reg664	-0.0554084	0.0391763	-1.4143342	0.1573677
reg665	0.0515041	0.0475598	1.0829330	0.2789253
reg666	0.0699968	0.0532960	1.3133585	0.1891628
reg667	0.0390596	0.0497416	0.7852502	0.4323690
reg668	-0.1980371	0.0525262	-3.7702521	0.0001662

effect	coef	std_error	t_stat	p_value
smsa66	0.0150626	0.0223322	0.6744764	0.5000606

```
two_stage$incorrectSE %>% knitr::kable(caption = "Incorrect Standard Errors")
```

Table 12: Incorrect Standard Errors

effect	coef	std_error	t_stat	p_value
Intercept	3.3396875	0.8805385	3.7927785	0.0001519
educ	0.1570593	0.0517554	3.0346457	0.0024289
exper	0.1188149	0.0224492	5.2926194	0.0000001
expersq	-0.0023565	0.0003421	-6.8887127	0.0000000
black	-0.1232778	0.0513339	-2.4014897	0.0163890
south	-0.1431945	0.0279996	-5.1141550	0.0000003
smsa	0.1007530	0.0310261	3.2473667	0.0011776
reg661	-0.1029760	0.0427428	-2.4091999	0.0160475
reg662	-0.0002287	0.0332654	-0.0068741	0.9945158
reg663	0.0469556	0.0321381	1.4610586	0.1441043
reg664	-0.0554084	0.0385696	-1.4365800	0.1509418
reg665	0.0515041	0.0468234	1.0999663	0.2714352
reg666	0.0699968	0.0524707	1.3340160	0.1823000
reg667	0.0390596	0.0489713	0.7976012	0.4251652
reg668	-0.1980371	0.0517128	-3.8295537	0.0001310
smsa66	0.0150626	0.0219864	0.6850851	0.4933432

(f) Hausman test

Should we worry about endogenaity? Conduct a Hausman test for endogeneity of educ. Report your test statistic, critical value and p-value.

Procedure:

1. Regress endogenous var X on instrument(s) Z. save residuals as v_hat
2. Include v_hat in original model
3. test if paramater coefficient on v-hat = 0 (ttest)

*Note: This test is only valid asymptotically (and, of course, is only as good as the instruments used).

```
Z_vars <- c("exper", "expersq", "black", "south", "smsa",
              "smsa66", "reg661", "reg662", "reg663", "reg664",
              "reg665", "reg666", "reg667", "reg668", "nearc4", "nearc2")

card_df %<>% mutate(v_hat = ols(card_df,"educ",Z_vars,T,F)$vars$e)

X_vars <- c("educ", "exper", "expersq", "black", "south", "smsa", "reg661", "reg662", "reg663", "r

orig <- ols(card_df,"lwage",X_vars,T,F,alpha = 0.05)
```

```r
orig$ttest
```

Table 13: OLS Results

	Coef.	S.E.	t Stat	p-Value	Decision
intercept	3.340	0.8214	4.066	0.0000	Reject
educ	0.157	0.0483	3.253	0.0012	Reject
exper	0.119	0.0209	5.673	0.0000	Reject
expersq	-0.002	0.0003	-7.384	0.0000	Reject
black	-0.123	0.0479	-2.574	0.0101	Reject
south	-0.143	0.0261	-5.482	0.0000	Reject
smsa	0.101	0.0289	3.481	0.0005	Reject
reg661	-0.103	0.0399	-2.583	0.0099	Reject
reg662	0.000	0.0310	-0.007	0.9941	Fail to Reject
reg663	0.047	0.0300	1.566	0.1174	Fail to Reject
reg664	-0.055	0.0360	-1.540	0.1237	Fail to Reject
reg665	0.052	0.0437	1.179	0.2384	Fail to Reject
reg666	0.070	0.0489	1.430	0.1528	Fail to Reject
reg667	0.039	0.0457	0.855	0.3926	Fail to Reject
reg668	-0.198	0.0482	-4.105	0.0000	Reject
smsa66	0.015	0.0205	0.734	0.4628	Fail to Reject
v_hat	-0.083	0.0484	-1.710	0.0873	Fail to Reject

The test statistic on v_hat is -1.710, corresponding to a p-value of 0.0873. The critical value for a 95% confidence level (significance level of 0.05) is $\pm$ 1.96, thus we fail to reject the null hypothesis, finding no significant evidence of endogeneity.