# ARE 212 Midterm

*Anaya Hall*

*March 14, 2018*

## Question One: "Linkages among climate change, crop yields and Mexico - US cross-border migration"

Load OLS functions

```
# Function to turn given data into matrix for use in OLS function
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)
}

# Function for OLS coefficient estimates and measures of fit
b_ols <- function(data, y_data, X_data, intercept=TRUE) {

  require(dplyr)
    # 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 beta hat ------
    b <- solve( t(X) %*% X ) %*% t(X) %*% y
    # Change the name of 'ones' to 'intercept'
    if(intercept == T){
        rownames(b) <- c("intercept", X_data) }
    else
        rownames(b) <- c(X_data)

    y_hat <- X %*% b
    e <- y - y_hat

    # Useful transformations -----
    n <- nrow(X) # number of observations
    k <- ncol(X) # number of independent variables
    dof <- n - k # degrees of freedom
```

```r
    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

    results <- data.frame(
      # The rows have the coef. names
      x_var = rownames(b),
      # Estimated coefficients
      coef = as.vector(b) %>% round(3)
    )

      # Return beta_hat & adjusted r2
    #return(R2adj)
    return(results)
}
```

```r
b_ols(data = auto, y_data = "price", X_data = c("mpg", "headroom"))
```

```
##       x_var      coef
## 1 intercept 12683.315
## 2       mpg  -259.106
## 3  headroom  -334.021
```

Load & clean data

1. **Estimate model (1) via OLS by regressing emigration rate on log of yields and a time period fixed effect.Report coefficient on yield and adjusted $R^2$. Does this match the results in the first column of table #1?**
2. Estimate model (1) again via fixed effects and FWT. Report coefficient on yield and adjusted $R^2$. Does this match the results in the third column of table #1?
3. Repeat step 1 without the the fixed effects. Report coefficient on yield and adjusted $R^2$. Do the results look different from what you estimated before? From what is in the paper?
4. Repeat step 2 without the the fixed effects. Report coefficient on yield and adjusted $R^2$. Do the results look different from what you estimated before? From what is in the paper?
5. What happened here? What are the consequences?

## Question Two: Normality of OLS

Model: $y_i = \beta_o + \beta_1 x_{1i} + \beta_2 x_{2i} + \epsilon_i$

Truth: $\beta_0$ = 3 , $\beta_1$ = 1, $\beta_2$ = -2

#### Load functions for use in simulation

Generate data function (given a sample size, n)

```
gen_data <- function(sample_size) {
  # Create data.frame with random x and error
  data_df <- data.frame(
    x1 = rnorm(sample_size),
    x2 = rnorm(sample_size),
    e = rnorm(sample_size))
  # Calculate y = 3 + 1 x1 - 2 x2 + e; drop 'e'
  data_df %<>% mutate(y = 3 + 1 * x1 - 2 * x2 + e) %>%
    select(-e)
  # Return data_df
  return(data_df)

}
```

Run a single simulation of OLS

```
one_sim <- function(sample_size) {
  # Estimate via OLS
  ols_est <- ols(data = gen_data(sample_size),
    y_data = "y", X_data = c("x1", "x2"))
  # Grab the estimated coefficient on x
  # (the second element of 'coef')
  b2 <- ols_est %$% coef[3]
  # Grab the second p-value
  # (the first p-value is for the intercept)
  # Return a data.frame with b1 and p_value
  return(data.frame(b1, p_value))
}
```