Problem Set 2

Andie Creel

2023-02-06

For both analyses this week, you will be using data from Mian and Sufi's 2014 Econometrica article, What Explains the 2007-2009 Drop in Employment?. The analyses will not match the exact numbers in the paper, but the full replication set is available if you are interested in exploring it.

Question One: Standard Errors

For this problem, use the dataset networth_delta_elas.csv, where county_fips is the county FIPS code, statename is the state FIPS code, elasticity is the Saiz elasticity measure, total is the number of households in each county, and netwp_h is the change in net worth within a county from 2006 to 2009.

Write a function to esitmate the linear regression of networth change against a constant and the Saiz elasticiy. Report the coefficient on the elasticity.

Let *netwp* be Y_i and *elasticity* be X_i where i indexes the county.

$$Y_i = \beta_0 + \beta_1 X_i$$

When finding $\hat{\beta}$ by minimizing the squared error,

$$\hat{\beta} = (X'X)^{-1}X'Y,$$

Note that X is a matrix and the first column is equal to 1 for the constant.

```
# -----------------------------------------------------------------
# read in data
# -----------------------------------------------------------------
# myData <- vroom('https://raw.githubusercontent.com/paulgp/applied-methods-phd/main/homework/Homework3
# vroom_write(myData, file = "networth_delta_elas.csv")

myData <- vroom('networth_delta_elas.csv')

# -----------------------------------------------------------------
# Function to estimate coefficients of OLS regressions
# -----------------------------------------------------------------
myOLS <- function(Y, X){
  beta <- solve(t(X)%*%X)%*%t(X)%*%Y

  return(beta)
}
```

```r
# ----------------------------------------------------------------
# Estimating effect of elasticity on network change
# ----------------------------------------------------------------
# getting matrices from data
myX <- cbind(1, myData$elasticity) #constant and elasticity
myY <- myData$netwp_h

myBeta <-myOLS(Y = myY, X = myX)


# Checking
#lm(data = myData, formula = netwp_h ~ elasticity)
```

I estimate β_0 is -0.1317989 and β_1 is 0.028727.

Next, estimate the homoskedastic SE, heteroskedasticity-robust SE, HC2, and HC3 standard errors for the elasticity estimate.

I use the code found on page 80 and 101 of *Econometrics* by Bruce Hansen (2014).

```r
# ----------------------------------------------------------------
# Set UP
# ----------------------------------------------------------------
x <- myX
y <- myY
e <- y - x%*%myBeta
#round(mean(e), 5)

n <- length(y)
k <- ncol(x)
a <- n/(n-k)
sig2 <- as.numeric((t(e) %*% e)/(n-k))
xx <- solve(t(x)%*%x) #X'X^-1

#Leverage
leverage <- rowSums(x*(x%*%solve(t(x)%*%x)))

# ----------------------------------------------------------------
# homoskedastic SE
# ----------------------------------------------------------------
v0 <- xx*sig2
s0 <- sqrt(diag(v0)) # Homoskedastic formula

# check
# summary(lm(data = myData, formula = netwp_h ~ elasticity))

# ----------------------------------------------------------------
# heteroskedasticity-robust SE
# ----------------------------------------------------------------
u1 <- x*(e%*%matrix(1,1,k))
v1 <- n/(n-k)*xx %*% (t(u1)%*%u1) %*% xx
s1 <- sqrt(diag(v1)) # Heteroskedastic-robust (White formula)
```

```
# -----------------------------------------------------------------
# HC2
# -----------------------------------------------------------------
u2 <- x*((e/sqrt(1-leverage))%*%matrix(1,1,k))
v2 <- xx %*% (t(u2)%*%u2) %*% xx
s2 <- sqrt(diag(v2)) # HC2 (Horn-Horn-Duncan formula)

# -----------------------------------------------------------------
# HC3
# -----------------------------------------------------------------
u3 <- x*((e/(1-leverage))%*%matrix(1,1,k))
v3 <- xx %*% (t(u3)%*%u3) %*% xx
s3 <- sqrt(diag(v3)) # HC3 (Andrews formula)
```

Homoskedastic SE: $SE_0 = 0.008177$, $SE_1 = 0.0033104$

Heteroskedasticity-Robust SE (aka Edgar Huber White formula): $SE_0 = 0.0111435$, $SE_1 = 0.004284$

HC2 (aka Horn-Horn-Duncan formula) SE: $SE_0 = 0.0114947$, $SE_1 = 0.0044938$

HC3 (aka Andrews formula): $SE_0 = 0.0119146$, $SE_1 = 0.0047362$

Now, we will estimate the three standard errors from Abadie et al. (2020) [see section 4]. I will walk you through the estimation.

$$V^{causal} = n^{-1}\Gamma^{-1}(\rho\Delta^{cond} + (1-\rho)\Delta^{ehw})\Gamma^{-1}$$
$$V^{causal,sample} = n^{-1}\Gamma^{-1}\Delta^{cond}\Gamma^{-1}$$
$$V^{descr} = n^{-1}(1-\rho)\Gamma^{-1}\Delta^{ehw}\Gamma^{-1}$$
$$V^{ehw} = n^{-1}\Gamma^{-1}\Delta^{ehw}\Gamma^{-1}$$

X is elasticity. Z is our constant. Y is the outcome of network change.

```
myFunction <- function(controls){
   # -----------------------------------------------------------------
  # Estimate e_hat as standard residual from Y ~ X + Z
  # -----------------------------------------------------------------
  n<- length(myData$elasticity)
  myZ <- controls #LITERALLY ONLY THING THAT'S CHANGED!
  myX <- myData$elasticity
  myY <- myData$netwp_h
  myZ <- as.matrix(myZ)

  myBeta <- myOLS(myY, cbind(myZ,myX))

  myStand_Resid <- myY - cbind(myZ,myX)%*%myBeta

  # -----------------------------------------------------------------
  # Estimate the short regression of X~Z to calculate gamma_hat,
  # which is the projection of X on Z. This is the mean when Z is a constant
  # -----------------------------------------------------------------
  myGamma_i <- myOLS(myX, myZ)
  myGamma_i <- as.numeric(myGamma_i)
```

```r
# Check
# mean(myX)

# --------------------------------------------------------------
# Estimate Gamma_hat
# --------------------------------------------------------------
gamma_sum <- rep(NA, n)

for (i in 1:n) {
  gamma_sum[i] <- (myX[i] - myGamma_i*myZ[i])^2
}

Gamma_hat <- 1/n * sum(gamma_sum)

# --------------------------------------------------------------
# Estimate Delta_ehw
# --------------------------------------------------------------
delta_sum <- rep(NA, n)

for (i in 1:n) {
  delta_sum[i] <- (myX[i] - myGamma_i*myZ[i,])*myStand_Resid[i]^2*(myX[i] - myGamma_i*myZ[i,])
}

Delta_ehw <- 1/n * sum(delta_sum) ; rm(delta_sum)


# --------------------------------------------------------------
# Now estimate V_EHW
# --------------------------------------------------------------
V_EHW <- (1/n)*Gamma_hat^{-1}*Delta_ehw*Gamma_hat^{-1} #ATTN: The check didn't match as close as I th
SE_EHW <- sqrt(V_EHW)

# --------------------------------------------------------------
# Estimate rho and V_descr
# --------------------------------------------------------------
myRho <- n/3006

V_descr <- (1-myRho)*V_EHW
SE_descr <- sqrt(V_descr)

# --------------------------------------------------------------
# Estimate G_hat
# --------------------------------------------------------------
G_sum_1 <- rep(NA, n)
G_sum_2 <- rep(NA, n)

for (i in 1:n) {
  G_sum_1[i] <- (myX[i] - myGamma_i%*%myZ[i,])%*%myStand_Resid[i]%*%t(myZ[i,])
  G_sum_2[i] <- myZ[i,] %*% t(myZ[i,])
}


G_hat <- (1/n*sum(G_sum_1))*(1/n*sum(G_sum_2))^{-1} ; rm(G_sum_1, G_sum_2)
```

```r
  # G_hat <- round(G_hat, digits = 14)

  # -------------------------------------------------------------
  # Estimate Delta_z
  # -------------------------------------------------------------
  Delta_sum <- rep(NA, n)

  for (i in 1:n) {
    Delta_sum[i] <- (myX[i] - myGamma_i%*%myZ[i,])%*%myStand_Resid[i] - G_hat*myZ[i,]
  }


  Delta_z <- 1/n*sum(Delta_sum^2)

  # Check
  # (round(Delta_z,15) == round(Delta_ehw, 15))

  # -------------------------------------------------------------
  # Estimate V_casaul, and V_causal_sample using Delta_z instead of Delta_cond
  # -------------------------------------------------------------
  V_causal <- 1/n * Gamma_hat^{-1}*(myRho*Delta_z+(1+myRho)*Delta_ehw)*Gamma_hat^{-1}
  SE_causal <- sqrt(V_causal)

  V_causal_sample <- 1/n*Gamma_hat^{-1}* Delta_z*Gamma_hat^{-1}
  SE_causal_sample <- sqrt(V_causal_sample)


  # -------------------------------------------------------------
  # Table
  # -------------------------------------------------------------
  myResults <- as.data.frame(cbind(Types = c('EHW', 'Descr', 'Causal', "Causal, Sample"),
                                   SE = c(SE_EHW, SE_descr, SE_causal, SE_causal_sample)))
}

myResults <- myFunction(controls = rep(1, n))


stargazer::stargazer(myResults, type = 'latex', summary = FALSE)
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Fri, Feb 10, 2023 - 14:10:48

Table 1:

	Types	SE
1	EHW	0.00427609297335225
2	Descr	0.00387301507438813
3	Causal	0.00498542092680068
4	Causal, Sample	0.00427609297335225

The EHW standard error is slightly larger than the Descr SE.

Reimplement this appraoch but include state fixed effects as controls in Z. Report your estimates for the standard errors using V^{EHW}, V^{descr}, V^{causal}, $V^{causal,sample}$.

```
# ---------------------------------------------------------------
# Create matrix Z for fixed effects
# ---------------------------------------------------------------
states_list <- unique(myData$statename)

# minus one to drop one fixed effect so that we are not over specified
myZ_fe <- matrix(nrow = n, ncol=length(states_list)-1)

for (i in 1:length(states_list)-1) {
  for (j in 1:n) {
    myZ_fe[j,i] <- (myData$statename[j] == states_list[i])
  }
}

myResults_2 <- myFunction(myZ_fe)
stargazer::stargazer(myResults_2, type = 'latex', summary = FALSE)
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Fri, Feb 10, 2023 - 14:10:48

Table 2:

	Types	SE
1	EHW	0.00073658306225617
2	Descr	0.000667150438831703
3	Causal	0.00081093867412994
4	Causal, Sample	0.000313024937590708

The EHW standard error is slightly larger than the Descr SE.

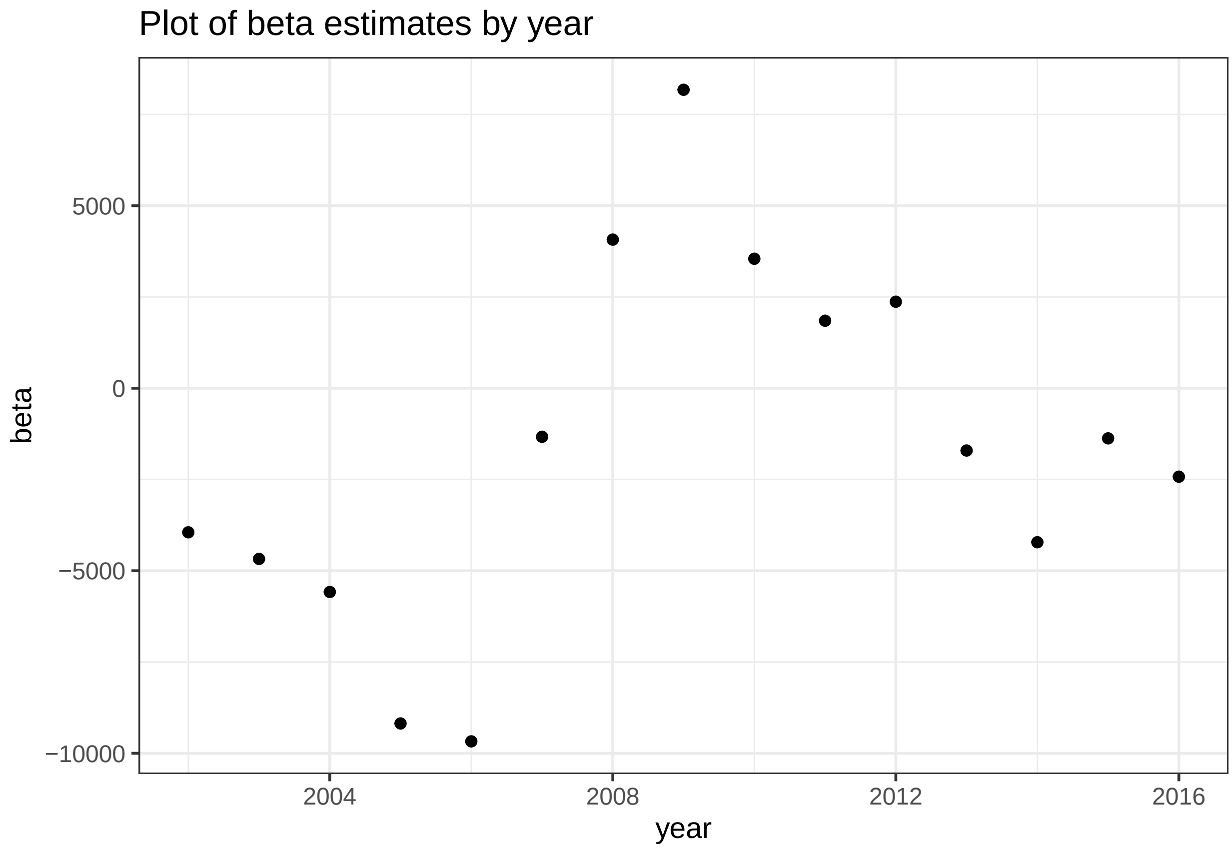
Question 2: Binscatter

For this problem, use the dataset networth_delta_elas.csv, where county fips is the county FIPS code, statename is the state FIPS code, year is the year, elasticity is the Saiz elasticity measure, total is the number of households in each county, and hpi is Zillow House Price value. For this problem, you may use your own regression estimate function, or a pre-existing function to estimate the regression.

Calculate annual house price appreciation (hpa) within each county, and regress HPA on the elasticity measure interacted with year, using your constructed function. I.e. $hpa_{it} = \alpha_t + \sum_s elasticity_i \times 1(year_t = S)\beta_t$. Plot the β_t coefficient for each year across time. Report the coefficient measuring the effect of elasticity in 2008.

```
# ---------------------------------------------------------------
# read in data
# ---------------------------------------------------------------
rm(list = ls())
```

```r
# myData <- vroom('https://raw.githubusercontent.com/paulgp/applied-methods-phd/main/homework/Homework3/
# vroom_write(myData, file = "yearly.csv")

myData <- vroom('yearly.csv') %>%
  mutate(year = as.factor(year))

# ---------------------------------------------------------------
# estimate regression
# ---------------------------------------------------------------
myData <- myData %>%
  group_by(county_fips) %>%
  mutate(hpa = hpi - lag(hpi))


myReg <- lm(data = myData, formula = hpa ~ year+ elasticity:year)
myCoefs <- myReg$coefficients[grepl('elasticity', names(myReg$coefficients))] # only the beta coefs

plotData <- as.data.frame(cbind(year = 2002:2016, beta = myCoefs))

# ---------------------------------------------------------------
# Plot
# ---------------------------------------------------------------
ggplot(data = plotData, aes(x = year, y=beta))+
  geom_point() +
  ggtitle("Plot of beta estimates by year") +
  theme_bw()
```



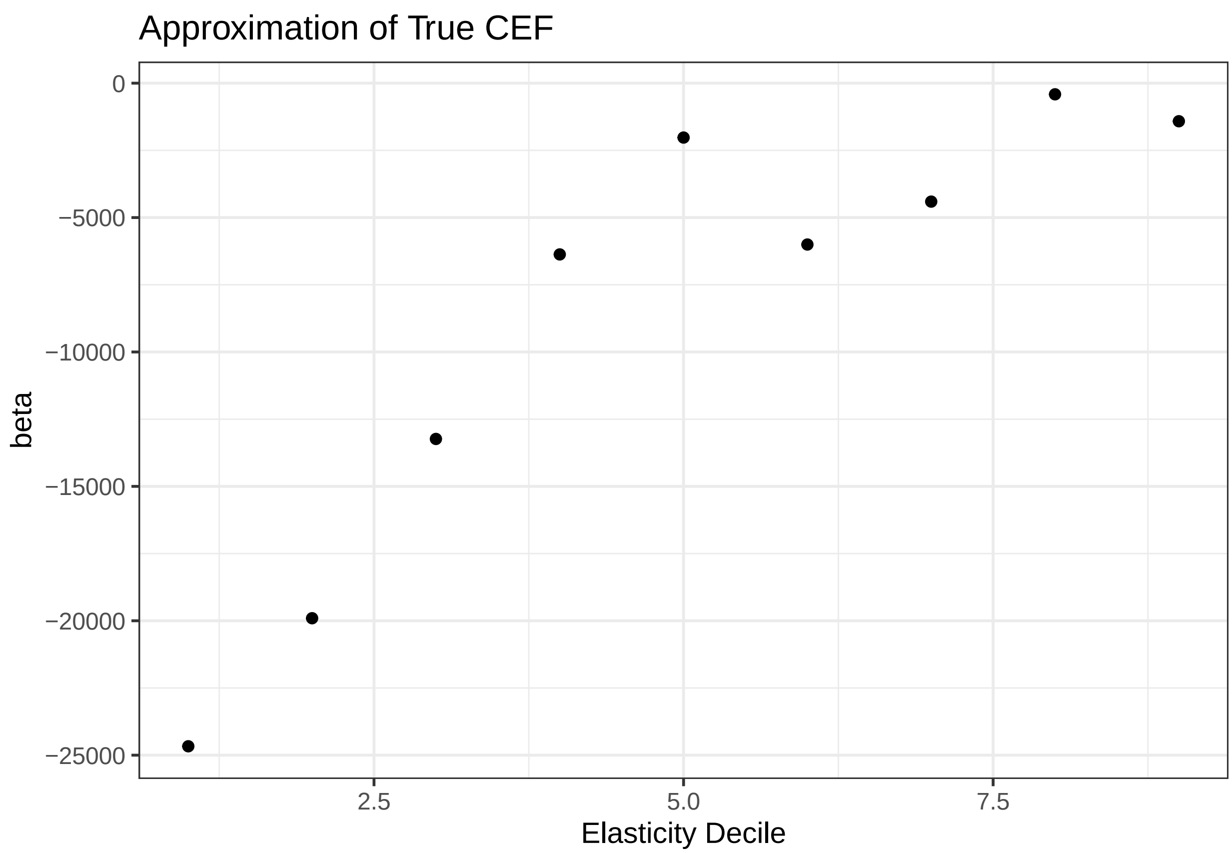
The β_{2008} coefficient is NA

Construct 10 decile dummies for the elasticity and reestimate the regression, pooling the years 2008-2010, and using the ten dummies in the place of the continuous elasticity measure. Plot these decile effects such that each point reflects an approximation to the conditional expectation function. Report the value for the first decile.

```r
# --------------------------------------------------------------
# data work
# --------------------------------------------------------------
#pooling data (??) I filter to 2008-2010
myData_b <- myData %>%
  ungroup() %>%
  filter(year == 2008 | year == 2009 | year == 2010) %>%
  mutate(decile = ntile(elasticity, 10)) %>% # create a decile dummy
  mutate(decile = as.factor(decile)) %>%
  mutate(decile = relevel(decile, ref = 10)) # using last decile as level

quantile(myData_b$elasticity, probs = seq(.1, .9, by = .1))
```

```
##      10%      20%      30%      40%      50%      60%      70%      80%
## 1.059162 1.456468 1.692800 2.112380 2.340009 2.553699 2.937962 3.392287
##      90%
## 4.003827
```

```r
# ------------------------------------------------------------------
# regress on decile dummies
# ------------------------------------------------------------------
myReg_b <- lm(myData_b, formula = hpa ~ decile)
myCoefs_b <- myReg_b$coefficients[2:10] # only the beta coefs

# ------------------------------------------------------------------
# Plot
# ------------------------------------------------------------------
plotData_b <- as.data.frame(cbind(elast_dec = 1:9, beta = myCoefs_b))

ggplot(data = plotData_b, aes(x = elast_dec, y=beta))+
  geom_point() +
  ggtitle("Approximation of True CEF") +
  xlab("Elasticity Decile") +
  theme_bw()
```



I find the coefficient for the first decile is -24668.6804355. This is in comparison to the 10th decile which is the level of my decile factor.

Citations

Hansen, Bruce. Econometrics. Princeton: Princeton University Press, 2014.