Problem Set 4

Prasanna Parasurama

Due: 3/29/19

Packages to Install

The packages used this week are

- estimatr (Tidyverse version of lm function)
- plm (Panel Data package)

Problem 1 (Analytical Exercise)

Assume we have the following dynamic panel model:

$$y_{i,t} = \gamma y_{i,t-1} + \alpha_i + \epsilon_{i,t},$$

We have the following assumptions:

$$\begin{aligned} |\gamma| &< 1 \\ y_{i,0} &\text{ is known (i.e. non-random)} \\ \mathbb{E}_T[\epsilon_{i,t}|y_{i,t-1}, ..., y_{i,0}, \alpha_i] &= 0 \text{ (Sequential Exogeneity)} \end{aligned}$$

Our goal is to consistently estimate γ.

1. Show that the first-difference estimator of γ is unbiased if $\alpha_i = 0$. (Hint: To show unbiased, show that the $\mathbb{E}[\hat{\gamma}_i] = \gamma i$.)

(Note: Since, $\epsilon_{i,t} \perp y_{i,t-1}$ (from exogenity assumption), when $\alpha = 0$, simple OLS estimator will be unbiased.)

Least Squares First-difference estimator is:

$$\hat{\gamma}_{FD} = \gamma + (\Delta y'_{i,t-1} \Delta y_{i,t-1})^{-1} (\Delta y'_{i,t-1} \Delta \epsilon_{i,t})$$

We need to show $(\Delta y'_{i,t-1} \Delta \epsilon_{i,t}) = 0$

$$(\Delta y'_{i,t-1} \Delta \epsilon_{i,t}) = (y_{i,t-1} - y_{i,t-2})'(\epsilon_{i,t} - \epsilon_{i,t-1})$$

$$\begin{aligned} \mathbb{E}\big[\mathbb{E}\big[(y_{i,t-1} - y_{i,t-2})'(\epsilon_{i,t} - \epsilon_{i,t-1})|y_{t-1}...y_0\big]\big] = E[\mathbb{E}\big[y_{i,t-1}\epsilon_{i,t}|y_{t-1}...y_0\big] + \\ \mathbb{E}\big[y_{i,t-2}\epsilon_{i,t}|y_{t-1}...y_0\big] + \\ \mathbb{E}\big[y_{i,t-1}\epsilon_{i,t-1}|y_{t-1}...y_0\big] + \\ \mathbb{E}\big[y_{i,t-2}\epsilon_{i,t-1}|y_{t-1}...y_0\big]\big] \end{aligned}$$

Note that all the terms are 0 due to the sequential exogenity assumption.

$$\mathbb{E}\big[y_{i,t-1}\epsilon_{i,t}|y_{t-1}...y_0\big] = \mathbb{E}\big[y_{i,t-1}\big]\mathbb{E}\big[\epsilon_{i,t}|y_{t-1}...y_0\big] = 0$$

$$\mathbb{E}\big[y_{i,t-2}\epsilon_{i,t}|y_{t-2}...y_0\big] = \mathbb{E}\big[y_{i,t-2}\big]\mathbb{E}\big[\epsilon_{i,t}|y_{t-1}...y_0\big] = 0$$

$$\mathbb{E}\big[y_{i,t-1}\epsilon_{i,t-1}|y_{t-1}...y_0\big] = \mathbb{E}\big[y_{i,t-1}\mathbb{E}\big[\epsilon_{i,t-1}|y_{t-1}...y_0\big]\big] = 0$$

$$\mathbb{E}\big[y_{i,t-2}\epsilon_{i,t-1}|y_{t-2}...y_0\big] = \mathbb{E}\big[y_{i,t-2}\big]\mathbb{E}\big[\epsilon_{i,t-1}|y_{t-1}...y_0\big] = 0$$

Hence, $\mathbb{E}\big[\hat{\gamma}_{FD}\big] = \gamma$

2. Is the first difference estimator of γ still unbiased if $\alpha_i \neq 0$? (Hint: Try showing that $cov(\epsilon_{i,t}, y_{i,t+1}) \neq 0$. Why is this condition useful?)

NO, We can follow the same exercise as (1), but the individual terms are not 0. For example,

$$\mathbb{E}\big[\epsilon_{i,t}|y_{t-1}...y_0\big] \neq 0$$

It is only 0 when α is known to satisfy the seq exogenity assumption.

$$E_T[\epsilon_{i,t}|y_{i,t-1}, ..., y_{i,0}, \alpha_i] = 0$$

3. Let $\Delta y_{i,t} = y_{i,t} - y_{i,t-1}$, what is $cov(\Delta\epsilon_{i,t}, y_{i,t-2})$? What is $cov(\Delta y_{i,t}, y_{i,t-2})$?

$$Cov(\Delta\epsilon_{i,t}, y_{i,t-2}) = \mathbb{E}\big[\Delta\epsilon_{i,t}y_{i,t-2}\big] - \mathbb{E}\big[\Delta\epsilon_{i,t}\big]\mathbb{E}\big[y_{i,t-2}\big]$$

By the seq exogenity assumption, $\Delta\epsilon_{i,t} \perp y_{i,t-2}$, which implies:

$$\mathbb{E}\big[\Delta\epsilon_{i,t}y_{i,t-2}\big] = \mathbb{E}\big[\Delta\epsilon_{i,t}\big]\mathbb{E}\big[y_{i,t-2}\big]$$

$$Cov(\Delta\epsilon_{i,t}, y_{i,t-2}) = \mathbb{E}\big[\Delta\epsilon_{i,t}\big]\mathbb{E}\big[y_{i,t-2}\big] - \mathbb{E}\big[\Delta\epsilon_{i,t}\big]\mathbb{E}\big[y_{i,t-2}\big] = 0$$

Similarly:

$$cov(\Delta y_{i,t}, y_{i,t-2}) = cov((y_{i,t}y_{i,t-1}), y_{i,t-2})$$

$y_{i,t-1} and y_{i,t-2}$ are clearly correlated (if $\gamma \neq 0$) my model construction, so:

$$cov(\Delta y_{i,t}, y_{i,t-2}) \neq 0$$

4. Using your answer from part (4), propose a strategy to consistently estimate γ? (Hint: The estimator will be a 2SLS (or IV regression).)

As seen above, $y_{i,t-2}$ satisfies both the relevance assumption ($cov(\Delta y_{i,t}, y_{i,t-2}) \neq 0$) and the exclusion restriction assumption($Cov(\Delta\epsilon_{i,t}, y_{i,t-2}) = 0$). So, $y_{i,t-2}$ can be used as an IV to estimate γ.

Problem 2 (Coding Exercise)

For this exercise we will be using data on cigarette sales available in the **plm** package. To load this dataset, simply use the following command:

```
library(plm)
library(estimatr)
library(dplyr)
library(AER)
data(Cigar,package="plm")
```

The data has the following columns:

  - **sales** – cigarette sales in packs per capita
  - **pimin** – minimum price in adjoining states per pack of cigarettes
  - **ndi** – per capita disposable income
  - **cpi** – consumer price index (1983=100)
  - **pop16** – population above the age of 16
  - **pop** – population
  - **price** – price per page of cigarettes
  - **year** – the year
  - **state** – number for the state

We will be estimating a series of panel models trying to estimate the price elasticity of demand.

1. The price elasticity of cigarette demand, E_d is the percentage change of sales of cigarettes divided by the percentage change of price of cigarettes. Let S_t be sales of cigarettes at time t and p_t be price of cigarettes at time t, then the price elasticity of demand is:

$$E_d = \frac{\Delta S_t/S_t}{\Delta p_t/p_t}$$

To estimate price elasticities in a linear regression, turn sales and price into logs. Why do we do this?

In a linear regression, logs captures the % change; the coefficients can be easily interpreted as elasticities.

```r
df = Cigar
df = mutate(df, log_sales=log(sales), log_price=log(price))
```

2. Assume that we are estimating the following cross-sectional regression:

$$log(S_{i,t}) = \gamma + \beta log(p_{i,t}) + \alpha_1 ndi_{i,t} + \alpha_2 cpi_{i,t} + \alpha_3 pop16_{i,t} + \epsilon_{i,t}$$

Estimate this regression, reporting the price elasticity of demand? What is potentially wrong with this regression?

```r
base_model = lm(log_sales~log_price+ndi+cpi+pop16, data=df)
summary(base_model)
```

```
## 
## Call:
## lm(formula = log_sales ~ log_price + ndi + cpi + pop16, data = df)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.74527 -0.10315  0.00986  0.09893  0.84256 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  7.410e+00  1.492e-01  49.681  < 2e-16 ***
## log_price   -8.386e-01  5.015e-02 -16.723  < 2e-16 ***
## ndi          3.188e-05  4.067e-06   7.839 9.07e-15 ***
## cpi          7.562e-03  7.969e-04   9.490  < 2e-16 ***
## pop16       -2.531e-06  1.545e-06  -1.638    0.102
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1987 on 1375 degrees of freedom
## Multiple R-squared:  0.2196, Adjusted R-squared:  0.2173
## F-statistic: 96.72 on 4 and 1375 DF,  p-value: < 2.2e-16
```

Price elasticity is -0.836.

Few problems with this regression: - Price and sales are jointly determined, so elasticity is likely to be biased (simultaneity bias). - Heterogenity across state, year etc

3. One important issue could be heterogeneity in sales across states, γ varies across states. However, we may not know whether we want to model this as a fixed effect or a random effect. One test often used is the Hausman-Wu test, available as phtest in the plm package. Estimate a within estimator, a random effect estimator and then use a Hausman-Wu test to report which specification is the correct one. Be sure to understand what the null and alternative hypothesis of this test are in order to report the correct interpretation of the test.

Random Effects:

```
state_random_effects = plm(log_sales~log_price+ndi+cpi+pop16, data=df, index=c("state"), model="random"
summary(state_random_effects)
```

```
## Oneway (individual) effect Random Effect Model
##    (Swamy-Arora's transformation)
##
## Call:
## plm(formula = log_sales ~ log_price + ndi + cpi + pop16, data = df,
##     effect = "individual", model = "random", index = c("state"))
##
## Balanced Panel: n = 46, T = 30, N = 1380
##
## Effects:
##                    var  std.dev share
## idiosyncratic 0.009871 0.099355 0.297
## individual    0.023316 0.152696 0.703
## theta: 0.882
##
## Residuals:
##       Min.    1st Qu.     Median    3rd Qu.       Max.
## -0.3802316 -0.0544012  0.0037906  0.0623338  0.4802717
##
## Coefficients:
##                Estimate  Std. Error  t-value Pr(>|t|)
## (Intercept)  6.0128e+00  1.0153e-01  59.2197  < 2e-16 ***
## log_price   -4.1110e-01  3.3040e-02 -12.4424  < 2e-16 ***
## ndi         -4.5424e-05  3.1747e-06 -14.3080  < 2e-16 ***
## cpi          1.0463e-02  5.2705e-04  19.8531  < 2e-16 ***
## pop16        6.3322e-06  3.5924e-06   1.7627  0.07817 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    23.146
## Residual Sum of Squares: 13.915
## R-Squared:      0.39883
## Adj. R-Squared: 0.39708
## F-statistic: 228.052 on 4 and 1375 DF, p-value: < 2.22e-16
```

Fixed Effects:

```
state_fixed_effects = plm(log_sales~log_price+ndi+cpi+pop16, data=df, index=c("state"), model="within",
summary(state_fixed_effects)
```

```
## Oneway (individual) effect Within Model
##
## Call:
## plm(formula = log_sales ~ log_price + ndi + cpi + pop16, data = df,
##     effect = "individual", model = "within", index = c("state"))
##
## Balanced Panel: n = 46, T = 30, N = 1380
##
## Residuals:
##       Min.    1st Qu.     Median    3rd Qu.       Max.
## -0.4133562 -0.0518554  0.0030902  0.0642572  0.4365386
##
## Coefficients:
##             Estimate  Std. Error  t-value Pr(>|t|)
## log_price -3.9847e-01  3.2803e-02 -12.1474  < 2e-16 ***
## ndi       -4.8093e-05  3.1669e-06 -15.1861  < 2e-16 ***
## cpi        1.0583e-02  5.2345e-04  20.2185  < 2e-16 ***
## pop16      7.7718e-06  4.2531e-06   1.8273  0.06788 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    22.491
## Residual Sum of Squares: 13.129
## R-Squared:      0.41625
## Adj. R-Squared: 0.39474
## F-statistic: 237.094 on 4 and 1330 DF, p-value: < 2.22e-16
```

Hausman-Wu Test:

```
phtest(state_random_effects, state_fixed_effects)
```

```
##
##  Hausman Test
##
## data:  log_sales ~ log_price + ndi + cpi + pop16
## chisq = 132.8, df = 4, p-value < 2.2e-16
## alternative hypothesis: one model is inconsistent
```

Null hypothesis is that both models are same, which can be rejected here.

4. We begin to believe that not only heterogeneity in mean sales, but dynamics are important, so we decide to estimate the following model:

$$log(S_{i,t}) = \gamma_i + \tau log(S_{i,t-1}) + \beta log(p_{i,t}) + \alpha_1 ndi_{i,t} + \alpha_2 cpi_{i,t} + \alpha_3 pop16_{i,t} + \epsilon_{i,t}$$

We are first going to have you implement the Anderson-Hsiao estimator:

5. Create a vector of first differences for each term in the regression above.

```
df = df %>% group_by(state) %>% mutate(log_sales_lag = lag(log_sales),
                                       log_sales_lag2 = lag(log_sales, 2),
                                       log_price_lag = lag(log_price),
                                       ndi_lag = lag(ndi),
                                       cpi_lag = lag(cpi),
                                       pop16_lag = lag(pop16))

df = df %>% mutate(sales_fd = log_sales - log_sales_lag,
                   sales_fd2 = log_sales_lag - log_sales_lag2,
                   price_fd = log_price - log_price_lag,
                   ndi_fd = ndi - ndi_lag,
                   cpi_fd = cpi - cpi_lag,
                   pop16_fd = pop16 - pop16_lag)
```

6. Let $\Delta S_{i,t} = log(S_{i,t}) - log(S_{i,t-1})$, we will be running the following two-stage least squares regression. In your first stage, use $log(S_{i,t-1})$ as an instrument for $\Delta S_{i,t}$, then run a 2SLS to get a new estimate for the price elasticity. Report any differences.

```
ah_model = ivreg(sales_fd ~ sales_fd2 + price_fd + ndi_fd + cpi_fd + pop16_fd | log_sales_lag2 + log_pr
summary(ah_model)
```

```
##
## Call:
## ivreg(formula = sales_fd ~ sales_fd2 + price_fd + ndi_fd + cpi_fd +
##     pop16_fd | log_sales_lag2 + log_price_lag + ndi_lag + cpi_lag +
##     pop16_lag, data = df)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.414071 -0.047462 -0.004556  0.040834  0.364237
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.576e-02  3.843e-02  -0.410   0.6818
## sales_fd2    1.605e+00  1.269e+00   1.265   0.2061
## price_fd    -5.159e-01  2.168e-01  -2.380   0.0175 *
## ndi_fd       3.280e-05  8.738e-05   0.375   0.7075
## cpi_fd       9.547e-03  7.993e-03   1.194   0.2325
## pop16_fd     1.365e-05  5.271e-05   0.259   0.7957
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08245 on 1282 degrees of freedom
## Multiple R-Squared: -2.515,  Adjusted R-squared: -2.529
## Wald test: 10.74 on 5 and 1282 DF,  p-value: 3.961e-10
```

Elasticity for FE was -0.4, with lag it's -0.5.

An issue with the Anderson-Hsiao estimator is that it is not efficient because it does not use all available information. The reason is quite simple, if $log(S_{i,t-1})$ is a valid instrument for $\Delta S_{i,t}$ and $log(S_{i,t-1})$ depends on $log(S_{i,t-2})$ then $log(S_{i,t-2})$ is also a valid instrument for $\Delta S_{i,t}$. This idea is the logic underpinning the Arellano-Bond estimator. This estimator is quite hard to code up, but luckily plm includes this estimator as the pgmm function.

7. Using a generalized method of moments estimator (Arellano-Bond), re-estimate the model. Report any differences.

```
ab_model = pgmm(sales_fd ~ sales_fd2 + price_fd + ndi_fd + cpi_fd + pop16_fd | lag(log_sales, 2:3), dat
```

```
## Warning in pgmm(sales_fd ~ sales_fd2 + price_fd + ndi_fd + cpi_fd +
## pop16_fd | : the first-step matrix is singular, a general inverse is used
```

```
## Error in solve.default(crossprod(WX, t(crossprod(WX, A1)))): system is computationally singular: rec
```