Panel Data 2: Implementation in R

Instructor: Yuta Toyama

Last updated: 2021-07-09

Panel

Preliminary:

- I use the following package
  - `lfe` package.

Panel Data Regression

- I use the dataset `Fatalities` in `AER` package.
  - See https://www.rdocumentation.org/packages/AER/versions/1.2-6/topics/Fatalities for details.

```
library(AER)
data(Fatalities)

str(Fatalities)
```

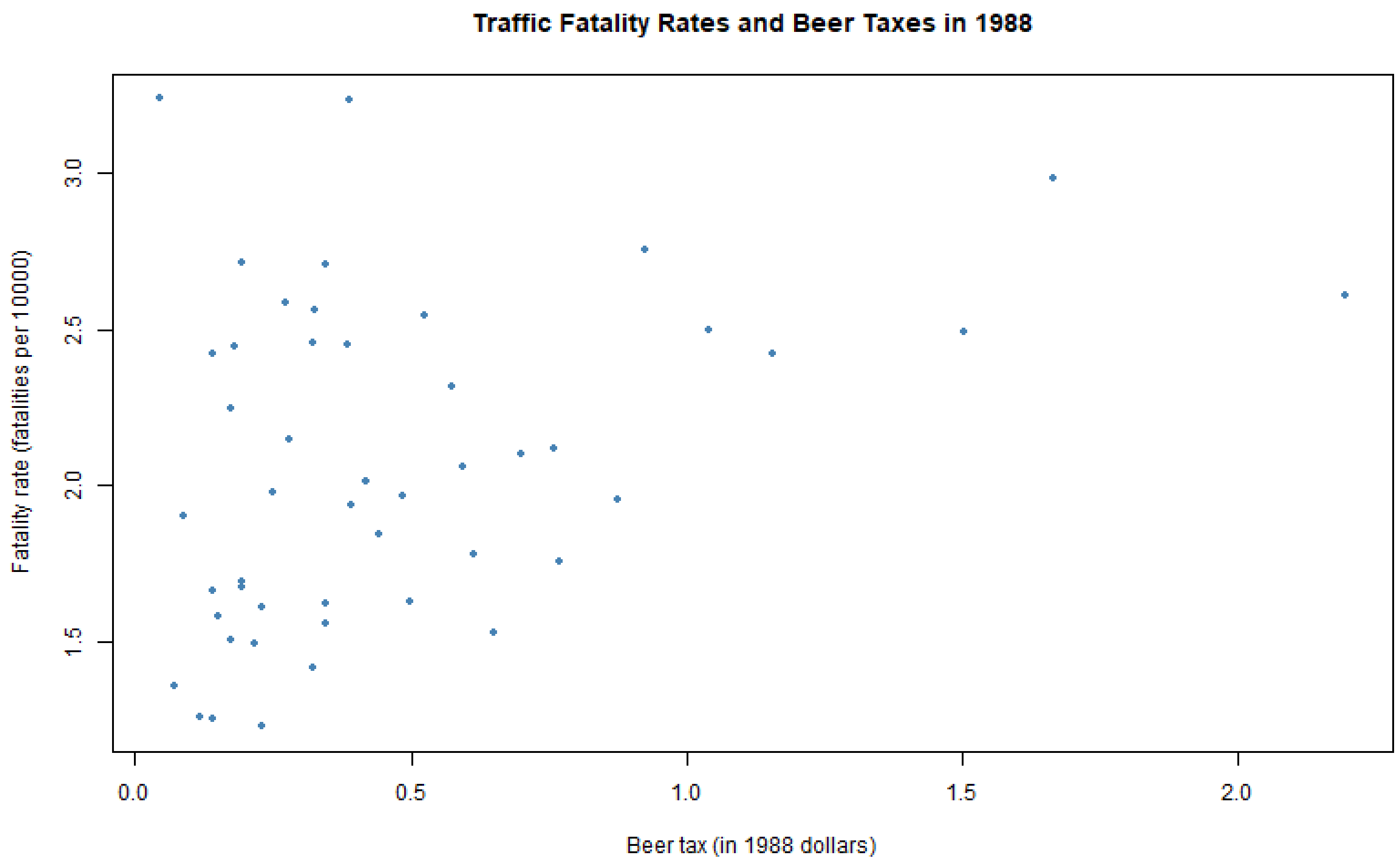
```
## 'data.frame':    336 obs. of  10 variables:
##  $ state   : Factor w/ 48 levels "al","az","ar",..: 1 1 1 1 1 1 1 2 2 2 ...
##  $ year    : Factor w/ 7 levels "1982","1983",..: 1 2 3 4 5 6 7 1 2 3 ...
##  $ spirits : num  1.37 1.36 1.32 1.28 1.23 ...
##  $ unemp   : num  14.4 13.7 11.1 8.9 9.8 ...
##  $ income  : num  10544 10733 11109 11333 11662 ...
##  $ emppop  : num  50.7 52.1 54.2 55.3 56.5 ...
##  $ beertax : num  1.54 1.79 1.71 1.65 1.61 ...
##  $ baptist : num  30.4 30.3 30.3 30.3 30.3 ...
##  $ mormon  : num  0.328 0.343 0.359 0.376 0.393 ...
##  $ drinkage: num  19 19 19 19.7 21 ...
```

```
## 'data.frame':    336 obs. of  24 variables:
##  $ dry         : num  25 23 24 23.6 23.5 ...
##  $ youngdrivers: num  0.212 0.211 0.211 0.211 0.213 ...
##  $ miles       : num  7234 7836 8263 8727 8953 ...
##  $ breath      : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ jail        : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 2 2 2 ...
##  $ service     : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 2 2 2 ...
##  $ fatal       : int  839 930 932 882 1081 1110 1023 724 675 869 ...
##  $ nfatal      : int  146 154 165 146 172 181 139 131 112 149 ...
##  $ sfatal      : int  99 98 94 98 119 114 89 76 60 81 ...
##  $ fatal1517   : int  53 71 49 66 82 94 66 40 40 51 ...
##  $ nfatal1517  : int  9 8 7 9 10 11 8 7 7 8 ...
##  $ fatal1820   : int  99 108 103 100 120 127 105 81 83 118 ...
##  $ nfatal1820  : int  34 26 25 23 23 31 24 16 19 34 ...
##  $ fatal2124   : int  120 124 118 114 119 138 123 96 80 123 ...
##  $ nfatal2124  : int  32 35 34 45 29 30 25 36 17 33 ...
##  $ afatal      : num  309 342 305 277 361 ...
##  $ pop         : num  3942002 3960008 3988992 4021008 4049994 ...
##  $ pop1517     : num  209000 202000 197000 195000 204000 ...
##  $ pop1820     : num  221553 219125 216724 214349 212000 ...
##  $ pop2124     : num  290000 290000 288000 284000 263000 ...
##  $ milestot    : num  28516 31032 32961 35091 36259 ...
##  $ unempus     : num  9.7 9.6 7.5 7.2 7 ...
##  $ emppopus    : num  57.8 57.9 59.5 60.1 60.7 ...
##  $ gsp         : num  -0.0221 0.0466 0.0628 0.0275 0.0321 ...
```

- As a preliminary analysis, let's plot the relationship between fatality rate and beer tax in 1998.

```
Fatalities %>%
  mutate(fatal_rate = fatal / pop * 10000) %>%
  filter(year == "1988") -> data
```

```
plot(x = data$beertax,
     y = data$fatal_rate,
     xlab = "Beer tax (in 1988 dollars)",
     ylab = "Fatality rate (fatalities per 10000)",
     main = "Traffic Fatality Rates and Beer Taxes in 1988",
     pch = 20,
     col = "steelblue")
```



- Positive correlation between alcohol tax and traffic accident. Possibly due to omitted variable bias.

- Run fixed effect regression using `felm` command in `lfe` package.
  - https://www.rdocumentation.org/packages/lfe/versions/2.8-3/topics/felm

```
library("lfe")

Fatalities %>%
    mutate(fatal_rate = fatal / pop * 10000) -> data

# OLS
result_ols <- felm( fatal_rate ~ beertax  | 0 | 0 | 0, data = data )
summary(result_ols, robust = TRUE)
```

```
##
## Call:
##    felm(formula = fatal_rate ~ beertax | 0 | 0 | 0, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.09060 -0.37768 -0.09436  0.28548  2.27643
##
## Coefficients:
##             Estimate Robust s.e t value Pr(>|t|)
## (Intercept)  1.85331    0.04713  39.324  < 2e-16 ***
## beertax      0.36461    0.05285   6.899 2.64e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5437 on 334 degrees of freedom
## Multiple R-squared(full model): 0.09336   Adjusted R-squared: 0.09065
## Multiple R-squared(proj model): 0.09336   Adjusted R-squared: 0.09065
## F-statistic(full model, *iid*):34.39 on 1 and 334 DF, p-value: 1.082e-08
## F-statistic(proj model): 47.59 on 1 and 334 DF, p-value: 2.643e-11
```

```r
# State FE
result_stateFE <- felm( fatal_rate ~ beertax  | state | 0 | state, data = data )
summary(result_stateFE, robust = TRUE)
```

```
##
## Call:
##    felm(formula = fatal_rate ~ beertax | state | 0 | state, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.58696 -0.08284 -0.00127  0.07955  0.89780
##
## Coefficients:
##         Estimate Cluster s.e. t value Pr(>|t|)
## beertax  -0.6559       0.2919  -2.247   0.0294 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1899 on 287 degrees of freedom
## Multiple R-squared(full model): 0.905   Adjusted R-squared: 0.8891
## Multiple R-squared(proj model): 0.04074   Adjusted R-squared: -0.1197
## F-statistic(full model, *iid*):56.97 on 48 and 287 DF, p-value: < 2.2e-16
## F-statistic(proj model):  5.05 on 1 and 47 DF, p-value: 0.02936
```

```r
# State and Year FE
result_bothFE <- felm( fatal_rate ~ beertax  | state + year | 0 | state, data = data )
summary(result_bothFE, robust = TRUE)
```

```
##
## Call:
##    felm(formula = fatal_rate ~ beertax | state + year | 0 | state,      data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.59556 -0.08096  0.00143  0.08234  0.83883
##
## Coefficients:
##         Estimate Cluster s.e. t value Pr(>|t|)
## beertax  -0.6400       0.3539  -1.809   0.0769 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1879 on 281 degrees of freedom
## Multiple R-squared(full model): 0.9089   Adjusted R-squared: 0.8914
## Multiple R-squared(proj model): 0.03606   Adjusted R-squared: -0.1492
## F-statistic(full model, *iid*):51.93 on 54 and 281 DF, p-value: < 2.2e-16
## F-statistic(proj model): 3.271 on 1 and 47 DF, p-value: 0.07692
```

Report results using texreg. Note that

- Setting "robust" option `TRUE` reports Heteroskedasticity-robust SE for the first column.
- Automatically report Cluster-Robust SE for the second and the third columns.

```
library(texreg)

screenreg(l = list(result_ols, result_stateFE, result_bothFE),
          digits = 3,
          # caption = 'title',
         # custom.model.names = c("(I)", "(II)", "(III)", "(IV)", "(V)"),
          custom.coef.names = NULL,  # add a class, if you want to change the names of variables.
          include.ci = F,
          include.rsquared = FALSE, include.adjrs = TRUE, include.nobs = TRUE,
          include.pvalues = FALSE, include.df = FALSE, include.rmse = FALSE,
          robust = T, # robust standard error
          custom.header = list("fatal_rate" = 1:3),
          custom.gof.rows = list("State FE"=c("No", "Yes", "Yes"), "Year FE"=c("No", "No", "Yes")),
          stars = numeric(0)
          )
```

```
##
## =====================================================
##                                 fatal_rate
##                        ----------------------------
##                        Model 1   Model 2   Model 3
## -----------------------------------------------------
## (Intercept)              1.853
##                         (0.047)
## beertax                  0.365    -0.656    -0.640
##                         (0.053)   (0.292)   (0.354)
## -----------------------------------------------------
## State FE                 No        Yes       Yes
## Year FE                  No        No        Yes
## Num. obs.               336       336       336
## Adj. R^2 (full model)    0.091     0.889     0.891
## Adj. R^2 (proj model)    0.091    -0.120    -0.149
## Num. groups: state                 48        48
## Num. groups: year                             7
## =====================================================
```

- What if we do not use the cluster-robust standard error?

```r
# State FE w.o. CRS
result_wo_CRS <- felm( fatal_rate ~ beertax  | state | 0 | 0, data = data )

# State FE w. CRS
result_w_CRS <- felm( fatal_rate ~ beertax  | state | 0 | state, data = data )

# Report heteroskedasticity robust standard error and cluster-robust standard errors
screenreg(l = list(result_wo_CRS, result_w_CRS),
          digits = 3,
          # caption = 'title',
         # custom.model.names = c("(I)", "(II)", "(III)", "(IV)", "(V)"),
          custom.coef.names = NULL,  # add a class, if you want to change the names of variables.
          stars = numeric(0),
          include.ci = F,
          include.rsquared = FALSE, include.adjrs = TRUE, include.nobs = TRUE,
          include.pvalues = FALSE, include.df = FALSE, include.rmse = FALSE,
          robust = T, # robust standard error
          custom.header = list("fatal_rate" = 1:2),
          custom.note = 'SE of `Model 1` is "Heteroskedasticity-Robust", while one of `Model 2` is '
          )
```

```
## 
## =========================================
##                             fatal_rate        
##                        -------------------
##                        Model 1   Model 2   
## -----------------------------------------
## beertax                 -0.656    -0.656   
##                         (0.203)   (0.292)  
## -----------------------------------------
## Num. obs.              336       336       
## Adj. R^2 (full model)    0.889     0.889   
## Adj. R^2 (proj model)   -0.120    -0.120   
## Num. groups: state      48        48       
## =========================================
## SE of `Model 1` is "Heteroskedasticity-Robust", while one of `Model 2` is "Cluster-Robust."
```

Panel + IV

Panel Data with Instrumental Variables

- Revisit the demand for Cigaretts
- Consider the following model

$$\log(Q_{it}) = \beta_0 + \beta_1 \log(P_{it}) + \beta_2 \log(income_{it}) + u_i + e_{it}$$

  where
  - Q_{it} is the number of packs per capita in state i in year t,
  - P_{it} is the after-tax average real price per pack of cigarettes, and
  - $income_{it}$ is the real income per capita. This is demand shifter.

- As an IV for the price, we use the followings:
  - $SalesTax_{it}$: the proportion of taxes on cigarettes arising from the general sales tax.
    - Relevant as it is included in the after-tax price
    - Exogenous(indepndent) since the sales tax does not influence demand directly, but indirectly through the price.
  - $CigTax_{it}$: the cigarett-specific taxes

```r
# load the data set and get an overview
library(AER)
data("CigarettesSW")
CigarettesSW %>%
  mutate( rincome = (income / population) / cpi) %>%
  mutate( rprice  = price / cpi ) %>%
  mutate( salestax = (taxs - tax) / cpi ) %>%
  mutate( cigtax = tax/cpi ) -> Cigdata
```

- Run IV regression with panel data.

```
# OLS
result_1 <- felm( log(packs) ~ log(rprice) + log(rincome)  | 0 | 0 | state, data = Cigdata )
# State FE
result_2 <- felm( log(packs) ~ log(rprice) + log(rincome)  | state | 0 | state, data = Cigdata )
# IV without FE
result_3 <- felm( log(packs) ~ log(rincome)  | 0 | (log(rprice) ~  salestax + cigtax) |
                      state, data = Cigdata )
# IV with FE
result_4 <- felm( log(packs) ~ log(rincome)  | state | (log(rprice) ~  salestax + cigtax) |
                      state, data = Cigdata )

screenreg(l = list(result_1, result_2, result_3, result_4),digits = 3,
          custom.coef.names = NULL,  # add a class, if you want to change the names of variables.
          include.ci = F,
          include.rsquared = FALSE, include.adjrs = TRUE, include.nobs = TRUE,
          include.pvalues = FALSE, include.df = FALSE, include.rmse = FALSE,
          robust = T, # robust standard error
          custom.header = list("log(packs)" = 1:4),
          stars = numeric(0))
```

```
##
## ==============================================================
##                                        log(packs)
##                         -------------------------------------
##                         Model 1  Model 2  Model 3  Model 4
## --------------------------------------------------------------
## (Intercept)             10.067            9.736
##                         (0.464)          (0.555)
## log(rprice)             -1.334   -1.210
##                         (0.174)  (0.143)
## log(rincome)             0.318    0.121    0.257    0.204
##                         (0.212)  (0.218)  (0.204)  (0.238)
## `log(rprice)(fit)`                        -1.229   -1.268
##                                           (0.183)  (0.162)
## --------------------------------------------------------------
## Num. obs.               96       96       96       96
## Adj. R^2 (full model)    0.542    0.929    0.539    0.929
## Adj. R^2 (proj model)    0.542    0.793    0.539    0.792
## Num. groups: state               48                48
## ==============================================================
```

`felm` command

Report heteroskedasticity robust standard error

```r
# Run felm command without specifying cluster.
result_1 <- felm( log(packs) ~ log(rprice) + log(rincome)  | 0 | 0 | state, data = Cigdata )


screenreg(l = list(result_1),
          digits = 3,
          # caption = 'title',
          custom.model.names = c("  Model 1  "),
          custom.coef.names = NULL,  # add a class, if you want to change the names of variables.
          include.ci = T,
          include.rsquared = FALSE, include.adjrs = TRUE, include.nobs = TRUE,
          include.pvalues = FALSE, include.df = FALSE, include.rmse = FALSE,
          robust = F, # robust standard error
          custom.header = list("log(packs)" = 1),
          stars = numeric(0),
          )


# stargazer::stargazer(result_1, type = "text",
#                      se = list(result_1$rse ) )
```

```
##
## ==================================
##                         log(packs)
##                        -----------
##                          Model 1
## ----------------------------------
## (Intercept)            10.067
##                        (0.464)
## log(rprice)            -1.334
##                        (0.174)
## log(rincome)            0.318
##                        (0.212)
## ----------------------------------
## Num. obs.              96
## Adj. R^2 (full model)   0.542
## Adj. R^2 (proj model)   0.542
## ==================================
```

How to conduct F test after `felm`

```
# Run felm command without specifying cluster.
result_1 <- felm( packs ~ rprice + rincome  | 0 | 0 | 0, data = Cigdata )


# The following tests H0: _b[rincome] = 0 & _b[rprice] = 0
ftest1 = waldtest(result_1, ~ rincome | rprice  )
ftest1
```

```
##            p         chi2          df1          p.F            F          df2
## 4.180596e-22 9.845284e+01 2.000000e+00 2.621701e-15 4.922642e+01 9.300000e+01
## attr(,"formula")
## ~rincome | rprice
## <environment: 0x000000003cb5bb40>
```

```
# ftest[5] corresponds to F-value
fval1 = ftest1[5]
fval1
```

```
##        F
## 49.22642
```

```r
# The following tests H0: _b[rincome] - 1 = 0 & _b[rprice] = 0
ftest2 = waldtest(result_1, ~ rincome - 1 | rprice  )
ftest2
```

```
##            p         chi2          df1          p.F            F          df2
## 2.048665e-24 1.090897e+02 2.000000e+00 2.121544e-16 5.454485e+01 9.300000e+01
## attr(,"formula")
## ~rincome - 1 | rprice
## <environment: 0x000000003cf70d08>
```

```r
# ftest[5] corresponds to F-value
fval2 = ftest1[5]
fval2
```

```
##        F
## 49.22642
```