

```

1  * To Run: do q4_code/pset4_q4.do
2  ****
3  * Problem Set 4, Question 4: CRE Event Study
4  * Students: Christina Brown, Sam Leone, Peter McCrory, Preston Mui
5  ****
6  qui{
7  set more off
8  // Specify Working Directory and import the data
9  global pers_dir =
10 "/Users/PBM/Dropbox/Berkeley/Fall_2016/Applied_Econometrics/"
11 global main = "$pers_dir/244-psets/ps4/"
12 cd "$main"
13 use curfews_class.dta, clear
14 }
15 *-----
16 * Part (a): Reshape the data into wide format
17 *-----
18 qui{
19 // Restrict the sample
20 keep if year >= 84 & year <= 90
21 // Reshape the Data
22 drop t
23 reshape wide lnarrests, i(city) j(year)
24 }
25 *-----
26 * Part (b): Estimate Seven Unrestricted Cross Sectional Regressions
27 *-----
28 qui{
29 // Dependent Variables
30 global depvars = "lnarrests84 lnarrests85 lnarrests86 lnarrests87 "
31 + ///
32 "lnarrests88 lnarrests89 lnarrests90"
33 // Indicator for Year Enacted
34 replace enacted = 999 if enacted > 90
35 qui tab enacted, gen(enacted_)
36 drop enacted_6
37 rename (enacted_1 enacted_2 enacted_3 enacted_4 enacted_5) ///
38 (enacted_85 enacted_87 enacted_88 enacted_89 enacted_90)
39
40 noisily mvreg $depvars = enacted_*

```

```

41 estimates store mvreg_est
42
43 // Export the table
44 do q4_code/export_table4b.do
45 }
46 *-----
47 * Part (c): Test the linear restrictions on  $\{\pi_{k,t}\}$  implied by
48 * (3)
49 *-----
50 qui{
51 global H "85 87 88 89 90"
52 // First set of linear restrictions, that for all t,  $t' < k$ ,
53  $\pi_{k,t} = \pi_{k,t'}$ 
54 foreach k in $H {
55     forvalues t = 84(1)90{
56         forvalues t_prime = 84(1)90 {
57             if `t' < `k' & `t_prime' < `k' & `t' < `t_prime'{
58                 *-----
59                 * Actual Test of First Set of Linear Restrictions
60                 *-----
61                 test _b[lnarrests`t':enacted_`k'] = ///
62                     _b[lnarrests`t_prime':enacted_`k'], `accum'
63                 *-----
64                 local accum = "accum"
65             }
66         }
67     }
68 // Second set of linear restrictions
69 foreach k in $H{
70     foreach k_prime in $H{
71         forvalues t = 84(1)90{
72             forvalues s = 84(1)90{
73                 if `k' < `k_prime' & `t' > `s' ///
74                 & `t' >= `k' & `s' >= `k' ///
75                 & `t' >= `k_prime' & `s' >= `k_prime'{
76                 *-----
77                 * Actual Test of Second Set of Linear Restrictions
78                 *-----
79                 qui test _b[lnarrests`t':enacted_`k_prime'] - _b[lnarrests`t':

```

```

enacted_`k`] ///
80     = ///
81     _b[lnarrests`s':enacted_`k_prime'] - _b[lnarrests`s':
enacted_`k'] ///
82     , accum
83 *-----
-----
84     }
85     }
86     }
87 }
88 }
89 // Third Set of Linear Restrictions
90 foreach k in $H{
91     foreach k_prime in $H{
92         forvalues t = 84(1)90{
93             forvalues s = 84(1)90{
94                 if `k' < `k_prime' & `t' > `s' ///
95                 & `t' >= `k' & `s' >= `k' ///
96                 & `t' >= `k_prime' & `s' >= `k_prime'{
97 *-----
-----
98 *           Actual Test of Third Set of Linear Restrictions
99 *-----
-----
100 qui test ///
101     _b[lnarrests`t':enacted_`k'] - _b[lnarrests`s':enacted_`k'] ///
102     = ///
103     _b[lnarrests`t':enacted_`k_prime'] - _b[lnarrests`s':enacted_
`k_prime'] ///
104     , accum
105 *-----
-----
106     }
107     }
108     }
109 }
110 }
111
112 // Fourth Set of Linear Restrictions
113 foreach k in $H{
114     foreach k_prime in $H{
115         forvalues t = 84(1)90{
116             forvalues s = 84(1)90{
117                 if `k' < `k_prime' ///
118                 & `t' < `k' & `s' < `k_prime'{
119                     forvalues j = 0(1)6{
120                         local k_for = `k' + `j'
121                         local k_prime_for = `k_prime' + `i'

```

```

121         local k_prime_for = k_prime_for
122         if `k_prime_for' < 90{
123 *-----
124 *               Actual Test of Fourth Set of Linear Restrictions
125 *-----
126 qui test ///
127     _b[lnarrests`k_for':enacted_`k'] - _b[lnarrests`t':enacted_`k']
128     ///
129     = ///
130     _b[lnarrests`k_prime_for':enacted_`k_prime'] ///
131     - _b[lnarrests`s':enacted_`k_prime'] ///
132     , accum
133 *-----
134     }
135 }
136 }
137 }
138 }
139 }
140 noisily test, accum
141 }
142 }
143
144 *-----
145 * Part (d): Constraint and Sureg
146 *-----
147 qui{
148 qui do q4_code/takefromglobal
149
150 local i = 1
151 global drop_constraints
152
153 while _rc == 0{
154     global drop_constraints $drop_constraints `r_dropped'
155     local r_dropped = r(dropped_`i')
156     local ++i
157     capture assert "`r(dropped_`i')'" != ""
158 }
159 global drop_constraints $drop_constraints `r_dropped'
160
161 constraint drop _all
162
163 local const num = 1

```

```

164
165 // First set of linear restrictions, that for all t, t' < k,
166 // pi_k,t = pi_k,t'
167 foreach k in $H {
168     forvalues t = 84(1)90{
169         forvalues t_prime = 84(1)90 {
170             if `t' < `k' & `t_prime' < `k' & `t' < `t_prime'{
171                 *-----
172                 *
173                 Specify the Constraint
174                 *-----
175
176                 constraint `const_num' ///
177                     [lnarrests`t']enacted_`k' = ///
178                     [lnarrests`t_prime']enacted_`k'
179                 *-----
180
181                 local ++const_num
182             }
183         }
184     }
185 }
186 // Second set of linear restrictions
187 foreach k in $H{
188     foreach k_prime in $H{
189         forvalues t = 84(1)90{
190             forvalues s = 84(1)90{
191                 if `k' < `k_prime' & `t' > `s' ///
192                 & `t' >= `k' & `s' >= `k' ///
193                 & `t' >= `k_prime' & `s' >= `k_prime'{
194                 *-----
195                 *
196                 Specify the Constraint
197                 *-----
198
199                 constraint `const_num' ///
200                     [lnarrests`t']enacted_`k_prime' - [lnarrests`t']enacted_`k' ///
201                     = ///
202                     [lnarrests`s']enacted_`k_prime' - [lnarrests`s']enacted_`k'
203                 *-----
204
205                 local ++const_num
206             }
207         }
208     }
209 }
210 // Third Set of Linear Restrictions

```

```

205 foreach k in $H{
206     foreach k_prime in $H{
207         forvalues t = 84(1)90{
208             forvalues s = 84(1)90{
209                 if `k' < `k_prime' & `t' > `s' ///
210                     & `t' >= `k' & `s' >= `k' ///
211                     & `t' >= `k_prime' & `s' >= `k_prime'{
212 *-----
213 *                Specify the Constraint
214 *-----
215 constraint `const_num' ///
216     [lnarrests`t']enacted_`k' - [lnarrests`s']enacted_`k' ///
217     = ///
218     [lnarrests`t']enacted_`k_prime' - [lnarrests`s']enacted_
`k_prime'
219 *-----
220 local ++const_num
221     }
222 }
223 }
224 }
225 }
226 // Fourth Set of Linear Restrictions
227 foreach k in $H{
228     foreach k_prime in $H{
229         forvalues t = 84(1)90{
230             forvalues s = 84(1)90{
231                 if `k' < `k_prime' ///
232                     & `t' < `k' & `s' < `k_prime'{
233                     forvalues j = 0(1)6{
234                         local k_for = `k' + `j'
235                         local k_prime_for = `k_prime' + `j'
236                         if `k_prime_for' < 90{
237 *-----
238 *                Specify the Constraint
239 *-----
240 constraint `const_num' ///
241     [lnarrests`k_for']enacted_`k' - [lnarrests`t']enacted_`k' ///
242     = ///
243     [lnarrests`k_prime_for']enacted_`k_prime' ///
244     - [lnarrests`s']enacted_`k_prime'
245 *-----

```

```

246 local ++const_num
247     }
248 }
249
250 }
251 }
252 }
253 }
254 }
255
256 local const_num = `const_num'-1
257
258 global constraints
259 forvalues j = 1(1)`const_num'{
260     global constraints "$constraints `j'"
261 }
262
263 takefromglobal constraints $drop_constraints
264
265 // Estimate the model
266 sureg (lnarrests84 enacted_*) (lnarrests85 enacted_*) ///
267       (lnarrests86 enacted_*) (lnarrests87 enacted_*) ///
268       (lnarrests88 enacted_*) (lnarrests89 enacted_*) ///
269       (lnarrests90 enacted_*), ///
270       constraints($constraints)
271 estimates store constrained_est
272
273 esttab constrained_est
274
275 // This file exports the table produced in part (b) of pset4_q4.do
276 esttab constrained_est using input/ps4_q4d.tex, ///
277     unstack se replace nonum ///
278     cells(b(star fmt(2)) se(par fmt(2))) ///
279     eqlabels("1984" "1985" "1986" "1987" "1988" "1989" "1990") ///
280     mlabels("Log of Arrests Made in", span ///
281             prefix(\multicolumn{@span}{c
282             }{) ///
283             suffix({})) ///
284     starlevels(* 0.10 ** 0.05 *** 0.01) ///
285     varlabels( _cons "Year Intercept" ///
286               enacted_84 "Enacted in 1984" ///
287               enacted_85 "Enacted in 1985" ///
288               enacted_86 "Enacted in 1986" ///
289               enacted_87 "Enacted in 1987" ///
290               enacted_88 "Enacted in 1988" ///
291               enacted_89 "Enacted in 1989" ///
292               enacted_90 "Enacted in 1990")
293 //Recover Estimates of Delta

```

```

293 //RECOVER ESTIMATES OF DELTA
294 matrix deltas = J(6,2,.)
295 forvalues i = 0(1)5{
296     local t = 85 + `i'
297     lincom [lnarrests`t']enacted_85-[lnarrests84]enacted_85
298
299     matrix deltas[`i'+1,1] = r(estimate)
300     matrix deltas[`i'+1,2] = r(se)
301 }
302 // Export the table
303 esttab matrix(deltas,fmt(3)) using input/ps4_q4d_lincoms.tex, ///
304     mlabel(,none) collabels("Estimate" "S.E.") tex replace ///
305     substitute(r1 "\$ \delta_0 = \pi_{85,85} - \pi_{85,84} \$" ///
306                r2 "\$ \delta_1 = \pi_{85,86} - \pi_{85,84} \$" ///
307                r3 "\$ \delta_2 = \pi_{85,87} - \pi_{85,84} \$" ///
308                r4 "\$ \delta_3 = \pi_{85,88} - \pi_{85,84} \$" ///
309                r5 "\$ \delta_4 = \pi_{85,89} - \pi_{85,84} \$" ///
310                r6 "\$ \delta_5 = \pi_{85,90} - \pi_{85,84} \$")
311 }
312
313
314
315

```