

```

1 use "C:\Users\yadto\Dropbox\USC\ECON513\HW4\Card-Krueger.dta", clear
2
3 quietly:{
4 noisily: display _n, _n, "-----question a -----"
5 // generate delta y
6 gen deltay = EMPL1 -EMPL0
7
8 //generate clusters by chains
9 gen bk = 0
10 replace bk = 1 if CHAIN == 1
11 gen kfc = 0
12 replace kfc = 1 if CHAIN == 2
13 gen roy = 0
14 replace roy = 1 if CHAIN == 3
15 gen wend = 0
16 replace wend = 1 if CHAIN ==4
17
18
19
20 // reg changeiny d bk kfc roy wend no intercept
21 noisily: reg deltay STATE bk kfc roy wend, nocon
22 noisily: display "coefficient of STATE is what we are interested "
23 noisily: display "compared to result in lecture 11, p9, beta is slightly bigger in
conditional "
24 noisily: display " --> dif-in-dif, and standard error seems to be no change."
25
26 //store the variance of coefficients
27 mat variance = get(VCE)
28 scalar varstate = variance[1,1]
29 scalar varbk = variance[2,2]
30 scalar varkfc = variance[3,3]
31 scalar varroy = variance[4,4]
32 scalar varwend = variance[5,5]
33 scalar sestate = sqrt(varstate)
34 scalar sebk = sqrt(varbk)
35 scalar sekfc = sqrt(varkfc)
36 scalar seroy = sqrt(varroy)
37 scalar sewend = sqrt(varwend)
38
39
40 noisily: display _n, _n, "-----question b-----"
41 //store the residuals
42 predict e, r
43
44 //construct matrix Z, indicator dtc, t time, c chain
45 gen d01 = 0
46 gen d02 = 0
47 gen d03 = 0
48 gen d04 = 0
49 gen d11 = 0
50 gen d12 = 0
51 gen d13 = 0
52 gen d14 = 0
53 replace d01 = 1 if STATE == 0 & bk == 1
54 replace d02 = 1 if STATE == 0 & kfc == 1
55 replace d03 = 1 if STATE == 0 & roy == 1
56 replace d04 = 1 if STATE == 0 & wend == 1
57 replace d11 = 1 if STATE == 1 & bk == 1
58 replace d12 = 1 if STATE == 1 & kfc == 1
59 replace d13 = 1 if STATE == 1 & roy == 1
60 replace d14 = 1 if STATE == 1 & wend == 1
61 mkmat d01 d02 d03 d04 d11 d12 d13 d14, mat(z) //this is matirx z
62 mkmat e, mat(e) // this is vector e
63
64 //number of observations in each cluster
65 forval i =1/4 {
66 qui sum d0`i'
67 qui gen n0`i' = r(sum)
68 qui sum d1`i'
69 qui gen n1`i' = r(sum)

```

```

70 }
71 noisily: sum n01 n02 n03 n04 n11 n12 n13 n14
72
73 // solve the etilde, sigma^2_tilde, thus rho
74 mata:
75 z = st_matrix("z")
76 e = st_matrix("e")
77 z2inv = invsym(z'*z)
78 I = I(391)
79 etilde = (I-z*z2inv*z')*e
80 variance = (etilde'*etilde)/(391-8-1)
81 st_matrix("var", variance)
82 end
83
84 scalar vartilde = var[1,1]
85 display vartilde //this is vartilde
86
87 sum e
88 scalar sdbar = r(sd)
89 scalar varbar = sdbar^2
90 display varbar //this is varhat
91
92 //it appears that varbar is smaller than vartilde, which is weird, I take the
93 // --> absolute value of their difference
94
95 scalar rhohat = (vartilde - varbar)/varbar
96 noisily: display "within cluster correlation:", rhohat //this is within cluster correlation
97 scalar varcluster = rhohat * varbar
98 noisily: display "variance of eta:", varcluster //this is variance of eta
99 scalar varindio = (1-rhohat)*varbar
100 noisily: display "variance of epsilon:", varindio //this is varince of epsilon
101
102 noisily: display _n,_n, "-----question c-----"
103
104 // correction factor
105 scalar L = 391/8
106 scalar correct = L*rhohat + (1-rhohat)
107 noisily: display "correction factor:", correct //this is correction factor
108
109 //corrected standard errors
110 scalar correctvarstate = correct*varstate
111 scalar correctsestate = sqrt(correctvarstate)
112 noisily: display "corrected standard error of state:", correctsestate //this is corrected
OLS standard error of STATE
113 scalar correctvarbk = correct*varbk
114 scalar correctsebk = sqrt(correctvarbk)
115 noisily: display "corrected standard error of bk:", correctsebk //this is corrected OLS
standard error of bk
116 scalar correctvarkfc = correct*varkfc
117 scalar correctsekfc = sqrt(correctvarkfc)
118 noisily: display "corrected standard error of kfc:", correctsekfc //this is corrected OLS
standard error of STATE
119 scalar correctvarroy = correct*varroy
120 scalar correctseroy = sqrt(correctvarroy)
121 noisily: display "corrected standard error of roy:", correctseroy //this is corrected OLS
standard error of STATE
122 scalar correctvarwend = correct*varwend
123 scalar correctsewend = sqrt(correctvarwend)
124 noisily: display "corrected standard error of wend:", correctsewend //this is corrected OLS
standard error of STATE
125
126
127 noisily: display _n,_n, "-----question d-----"
128
129 gen group = 0
130 replace group = 1 if STATE == 0 & bk == 1
131 replace group = 2 if STATE == 0 & kfc == 1
132 replace group = 3 if STATE == 0 & roy == 1
133 replace group = 4 if STATE == 0 & wend == 1
134 replace group = 5 if STATE == 1 & bk == 1

```

```

135 replace group = 6 if STATE == 1 & kfc == 1
136 replace group = 7 if STATE == 1 & roy == 1
137 replace group = 8 if STATE == 1 & wend == 1
138 sort group
139 by group: gen counter = _n
140 mkmat STATE bk kfc roy wend if counter ==1, mat(X)
141 mat list X //this is matrix X
142 mat X1 = (0\1\0\0\0)
143 mat X2 = (0\0\1\0\0)
144 mat X3 = (0\0\0\1\0)
145 mat X4 = (0\0\0\0\1)
146 mat X5 = (1\1\0\0\0)
147 mat X6 = (1\0\1\0\0)
148 mat X7 = (1\0\0\1\0)
149 mat X8 = (1\0\0\0\1)
150 scalar n1 = n01
151 scalar n2 = n02
152 scalar n3 = n03
153 scalar n4 = n04
154 scalar n5 = n11
155 scalar n6 = n12
156 scalar n7 = n13
157 scalar n8 = n14
158
159 mat sumM = J(5,5,.)
160 mat sumN = J(5,5,.)
161 forval i =1/8{
162 mat M`i' = n`i'*X`i'*X`i'
163 scalar n`i'sq = n`i'^2
164 mat N`i' = n`i'sq*(varcluster + varindio/n`i')*X`i'*X`i'
165 }
166
167 mat sumM = M1+M2+M3+M4+M5+M6+M7+M8
168 mat sumN = N1+N2+N3+N4+N5+N6+N7+N8
169 mat sumMinv = inv(sumM)
170 mat varbeta = sumMinv*sumN*sumMinv
171 scalar varbetastate = varbeta[1,1]
172 scalar varbetabk = varbeta[2,2]
173 scalar varbetakfc = varbeta[3,3]
174 scalar varbetaroy = varbeta[4,4]
175 scalar varbetawend = varbeta[5,5]
176 scalar sebetastate = sqrt(varbetastate)
177 scalar sebetabk = sqrt(varbetabk)
178 scalar sebetakfc = sqrt(varbetakfc)
179 scalar sebetaroy = sqrt(varbetaroy)
180 scalar sebetawend = sqrt(varbetawend)
181
182 noisily: display _n, "state bk kfc roy wend"
183 noisily: display "(OLS)", sestate, sebk, sekfc, seroy, sewend
184 noisily: display "(Correct)", sebetastate, sebetabk, sebetakfc, sebetaroy,
sebetawend
185 noisily: display "(Correction factor)", correctsestate, correctsebk, correctsekfc,
correctseroy, correctsewend
186
187
188
189 noisily: display _n, _n, "-----question e -----"
190 gen sbk = STATE*bk
191 gen skfc = STATE*kfc
192 gen sroy = STATE*roy
193 gen swend = STATE*wend
194 noisily: reg deltax STATE bk kfc roy wend sbk skfc sroy swend, nocon
195 noisily: display " we cannot do clustered standard errors since ols residual e has
properties:"
196 noisily: display " --> e*s*c has expectation of 0, where d is state, c is chain,"
197 noisily: display " --> so ols estimator of variance of cluster specific error is also 0."
198
199
200 noisily: display _n, _n, "-----question f -----"
201

```

```
202   noisily: reg deltax STATE bk kfc roy wend, nocon vce(boot)
203   noisily: display "the standard error grows bigger compared to a"
204
205   }
206
```