```
1   ****************************************
2   * PS 3 675 metrics
3   *Nate Mather
4   *Stata section
5   ****************************************
6
7   clear
8   **************
9   * Question 1 *
10  **************
11  * load data
12  use "C:\Users\Nmath 000\Documents\MI school\Second Year\675 Applied
    Econometrics\hw\hw3\pisofirme.dta",clear
13
14  * set wd
15  cd "C:\Users\Nmath_000\Documents\Code\courses\econ 675\PS_3_tex"
16
17  gen s = 1-dmissing
18  gen log_inc = ln(S_incomepc + 1)
19
20  *******
21  *part1*
22  *******
23
24  logit s S_age S_HHpeople log_inc, vce(robust)
25
26  * output for LaTeX
27  outreg2 using stata_tab_q1_9_a.tex, side stats(coef se tstat pval ci) ///
28   noaster noparen nor2 noobs dec(3) replace tex(frag)
29
30   *******
31  *part 2*
32  ********
33
34  * nonparametric bootstrap
35  logit s S_age S_HHpeople log_inc, vce(bootstrap, reps(999))
36
37  * output for LaTeX
38  outreg2 using stata_table_q1_9_b.tex, side stats(coef se tstat pval ci) ///
39   noaster noparen nor2 noobs dec(3) replace  tex(frag)
40
41  * Q1.9c - propensity scores
42  * logit regression, robust standard errors
43  logit s S_age S_HHpeople log_inc, vce(robust)
44
45  * predict propensity score
46  predict p
47
48  * plot histogram, overlay kernel density
49  twoway histogram p || kdensity p, k(gaussian)
50
51
52  * save
53  graph export stata_plot_q1_9_c.png, replace
54
55
56
57  ************************************************
58  **** Question 2
59
60  ************************************************
61
62
63  * gmm, four moment conditions
64  local vars = "dpisofirme S_age S_HHpeople log_inc"
65  gmm ((danemia - invlogit((dpisofirme*{theta}+S_age*{gamma1}+S_HHpeople*{gamma2}+log_inc*{
    gamma3})))*dpisofirme) ///
66  ((danemia - invlogit((dpisofirme*{theta}+S age*{gamma1}+S HHpeople*{gamma2}+log inc*{gamma3
    }))))*S_age) ///
67  ((danemia - invlogit((dpisofirme*{theta}+S_age*{gamma1}+S_HHpeople*{gamma2}+log_inc*{gamma3
```

```
    }))))*S_HHpeople) ///
68  ((danemia - invlogit((dpisofirme*{theta}+S_age*{gamma1}+S_HHpeople*{gamma2}+log_inc*{gamma3
    }))))*log_inc), ///
69  instruments(`vars') winitial(identity) vce(boot)
70
71  * output for LaTeX
72   mata:
73      coef = st_matrix("e(b)")'
74      se = st_matrix("e(se)")'
75
76      tstat = coef:/se
77
78      CI_low = coef - 1.96:*se
79      CI_high = coef + 1.96:*se
80
81      stats = round((coef,se,tstat,CI_low,CI_high),.001)
82
83      st_matrix("stats",stats)
84  end
85  mat rownames stats = `vars'
86  mat colnames stats = coef se tstat CI_low CI_high
87  outtable using stata_table_q2_2_b, mat(stats) replace nobox
88
89  * Q2.3c (MAR)- feasible estimator
90  * we predicted p before, but did not use t, so do that now:
91  * logit regression, robust standard errors
92  logit s dpisofirme S_age S_HHpeople log_inc, vce(robust)
93
94  * predict propensity score
95  predict p_witht
96
97  * now run gmm adding in new term s/p
98  local vars = "dpisofirme S_age S_HHpeople log_inc"
99  gmm ((s/p_witht)*(danemia - invlogit((dpisofirme*{theta}+S_age*{gamma1}+S_HHpeople*{gamma2}+
    log_inc*{gamma3}))))*dpisofirme) ///
100 ((s/p_witht)*(danemia - invlogit((dpisofirme*{theta}+S_age*{gamma1}+S_HHpeople*{gamma2}+
    log inc*{gamma3}))))*S age) ///
101 ((s/p_witht)*(danemia - invlogit((dpisofirme*{theta}+S_age*{gamma1}+S_HHpeople*{gamma2}+
    log_inc*{gamma3}))))*S_HHpeople) ///
102 ((s/p_witht)*(danemia - invlogit((dpisofirme*{theta}+S_age*{gamma1}+S_HHpeople*{gamma2}+
    log_inc*{gamma3}))))*log_inc), ///
103 instruments(`vars') winitial(identity) vce(boot)
104
105 * output for LaTeX
106  mata:
107     coef = st_matrix("e(b)")'
108     se = st_matrix("e(se)")'
109
110     tstat = coef:/se
111
112     CI_low = coef - 1.96:*se
113     CI_high = coef + 1.96:*se
114
115     stats = round((coef,se,tstat,CI_low,CI_high),.001)
116
117     st_matrix("stats",stats)
118 end
119 mat rownames stats = `vars'
120 mat colnames stats = coef se tstat CI_low CI_high
121 outtable using stata_table_q2_3_c, mat(stats) replace nobox
122
123 * Q2.3d (MAR)- feasible estimator, trimmed
124 * we predicted p before, and have s, so add that before the moment conditions
125 local vars = "dpisofirme S_age S_HHpeople log_inc"
126 gmm ((s/p_witht)*(danemia - invlogit((dpisofirme*{theta}+S_age*{gamma1}+S_HHpeople*{gamma2}+
    log_inc*{gamma3}))))*dpisofirme) ///
127 ((s/p_witht)*(danemia - invlogit((dpisofirme*{theta}+S_age*{gamma1}+S_HHpeople*{gamma2}+
    log inc*{gamma3}))))*S age) ///
128 ((s/p_witht)*(danemia - invlogit((dpisofirme*{theta}+S_age*{gamma1}+S_HHpeople*{gamma2}+
    log_inc*{gamma3}))))*S_HHpeople) ///
```

```
((s/p_witht)*(danemia - invlogit((dpisofirme*{theta}+S_age*{gamma1}+S_HHpeople*{gamma2}+
log_inc*{gamma3})))*log_inc) ///
if p_witht >= 0.1, instruments(`vars') winitial(identity) vce(boot)

* output for LaTeX
 mata:
    coef = st_matrix("e(b)")'
    se = st_matrix("e(se)")'

    tstat = coef:/se

    CI_low = coef - 1.96:*se
    CI_high = coef + 1.96:*se

    stats = round((coef,se,tstat,CI_low,CI_high),.001)

    st_matrix("stats",stats)
end
mat rownames stats = `vars'
mat colnames stats = coef se tstat CI_low CI_high
outtable using stata_table_q2_3_d, mat(stats) replace nobox









*******************************
*** Question 3
*******************************

* Q3.1 - nonparametric bootstrap
clear all

* generate sample
set seed 123
set obs 1000
gen X = runiform()

* save actual max
sum X
local maxX=r(max)

* run nonparametric bootstrap of max
bootstrap stat=r(max), reps(599) saving(nonpar_results, replace): summarize X

* load results
use nonpar_results, clear

* generate statistic
gen nonpar_stat = 1000*(`maxX'-stat)

* plot
hist nonpar_stat, ///
 plot(function exponential = 1-exponential(1,x), range(0 5) color(red))
graph export stata_plot_q3_1.png, replace

*******************************************************************************
* Q3.2 - parametric bootstrap
clear all

tempname memhold
tempfile para_results

* generate sample
set seed 123
set obs 1000
gen X = runiform()

* save actual max
```

```
198   sum X
199   local maxX=r(max)
200
201   * parametric bootstrap
202   postfile `memhold' max using `para_results'
203   forvalues i = 1/599{
204       capture drop sample
205       gen sample = runiform(0,`maxX')
206       sum sample
207       post `memhold' (r(max))
208   }
209   postclose `memhold'
210
211   * load results
212   use `para_results', clear
213
214   * generate statistic
215   gen para stat = 1000*(`maxX'-max)
216
217   * plot
218   hist para_stat, ///
219    plot(function exponential = 1-exponential(1,x), range(0 5) color(red))
220   graph export stata_plot_q3_2.png, replace
221
```