

PS 7

Question 1

b) The dist. of birth weights is not very smooth. In particular, the following two characteristics stand out:

i) From the 1-gram-bin hist., it's clear that a lot of obs. are concentrated on some specific gram-values. As it turns out, these focal points are gram-quantities that are multiples of 10, or gram-quantities (rounded) that correspond to measurements in ounces.

ii) From the 25-gram-bin-histogram, it's clear that there are "surprisingly" few obs. in the bin with range 1475-1499 grams.

c) We would have to be concerned that our running variable is manipulable, i.e., that there is selection towards one side of the cutoff. In the present case, it would be meaningful that "ambitious" / caring parents would try to downreport the weight of their child in order to get extra care. If there is bunching in the data, then it would be on the left side of the 1500-gram-cutoff.

However, and more plausible, it seems that nurses and doctors intentionally misreport around the cutoff by reporting 1500+ weights when the child in fact is below 1500g, in order not to have to give the extra treatment.

Related to manipulation is local independence, where we would test independence of baseline characteristics, which would have to hold for the RDD to be valid. This will be done in e).

d)

From the Stata code, I get the following jumps at the cutoff (jump from the left to the right)

	bwidth150	bwidth100	bwidth50
bin 1gr	447,1754	740,1982	1611,6
bin 10gr	4279,233	735,79	11093,7
bin 25gr	13216,73	20013	38534,5

none is significant

Why robust SEs

The dataset is such that higher birthweights are more frequent. Thus, it is natural to assume that there is heteroskedasticity, since frequencies are non-negative (compared to the wages-expl. where we have heterosk. by construction).

Even though there seems to be a jump in all specifications, the jumps are not significant, thus I would say we cannot reject that it's not smooth (meaning that implementing an RDD seems legit).

2) I used the rd command throughout the whole exercise, in i)-iii), rectangular kernel is added as an option, whereas for iv)-vi) it's the default option (triangular kernel).

Table: Results "jump" at cutoff

		mom-white	mom-ed1
kernel: rectangular	bw=90	0.0005634 (0.0115876)	0.0035626 (0.0068183)
	bw=60	0.0056396 (0.0151777)	0.0075472 (0.008805)
	bw=30	0.0124496 (0.01982)	-0.0066291 (0.034935)
kernel: triangular	bw=90	0.0048244 (0.0132902)	0.0047352 (0.0079835)
	bw=60	0.0084344 (0.0152047)	0.0019931 (0.009288)
	bw=30	0.0071293 (0.0200868)	-0.0009516 (0.0122499)

The estimates are all very small in magnitude, and not significant.
I would conclude that the RD design passes the balanced covariates test.

f)

When considering the outcome dummy white yes/no, most cune heaps as well as the 1500-gr-heaps feature significantly different outcomes.

With mom-ed1, i.e. dummy "less than highschool" yes/no, this is less pronounced; it seems to be mainly the 1500-gr-heaps that have a significant impact on mom-ed1.

g)

I repeat all of the $3 \cdot 2 = 6$ specifications from e), for both robust SEs / cluster SEs.
bandwidth: 30, 60, 90 kernel: rect./triang.

⇒ Effect on 1-yr.-mortality:

While all estimates are significant ("positive" jump from left to right side of cutoff, i.e. positive effect of "extra care"-treatment), they differ substantially, from 0.0074272 (for bw=90, kernel=rect.) to 0.026285 (for bw=30, kernel=triangular).

This might be evidence that est. is quite sensitive to specification.

Furthermore, SE are considerably bigger when clustering on grams. This suggests that there are 'independent clusters', and that one should use clustered SEs (on grams).

→ Hospitals in poorer areas measure less precisely, are more inclined to round to certain numbers (as multiples of 10), or simply have a preference for, say, reporting multiples of 10.

If this is correlated with general health conditions of the children (would expect the children to be less healthy in less privileged areas), and that there are many obs. from the same hospitals, then these are not independent observations. → clustered SEs.

i.) Code repeats all steps of g), dropping 1500gr.

→ When dropping the 1500s, the RD estimates are considerably lower for all specifications.

No, RD should never be sensitive to dropping obs. exactly at the cutoff, since this would suggest that there is bunching (non-random sorting). Since the weight scale is "finely granulated", the RD estimate for $1499 \leftrightarrow 1500$ should hardly be different from the RD estimate, $1499 \leftrightarrow 1501$, but yet, this seems to be the case here.

When dropping the obs. measured in ounces, the RD estimates are much larger, in fact as large as 0.0473915 (from left to right of cutoff) when using a 30-bandwidth, and triangular kernel.

Note: About the kernel, triangular seems to be better for minimizing mean squared error, and it is intuitively appealing (more weight for closer obs.). Further, I would cluster on bweight.

⇒ Main Spec.: bandwidth 60, triangular, drop if bweight = 1500, cluster (bweight), see regression fct. for specification.

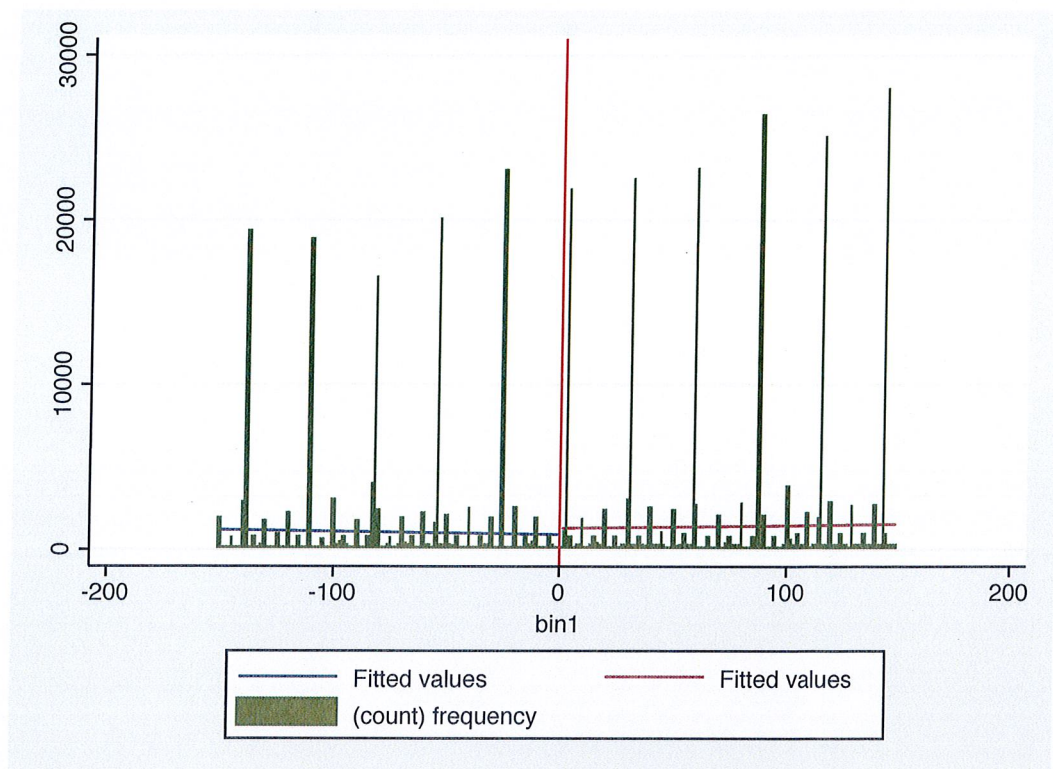
I conclude that the effect of very low bweight classification has a decreasing / negative effect on 1-year-mortality of ~ 1.012 pp. (which is a relatively large effect).

Considering the other estimates of the alternative specifications, this seems almost a conservative estimate - by the arguments highlighted before, I would trust it the most.

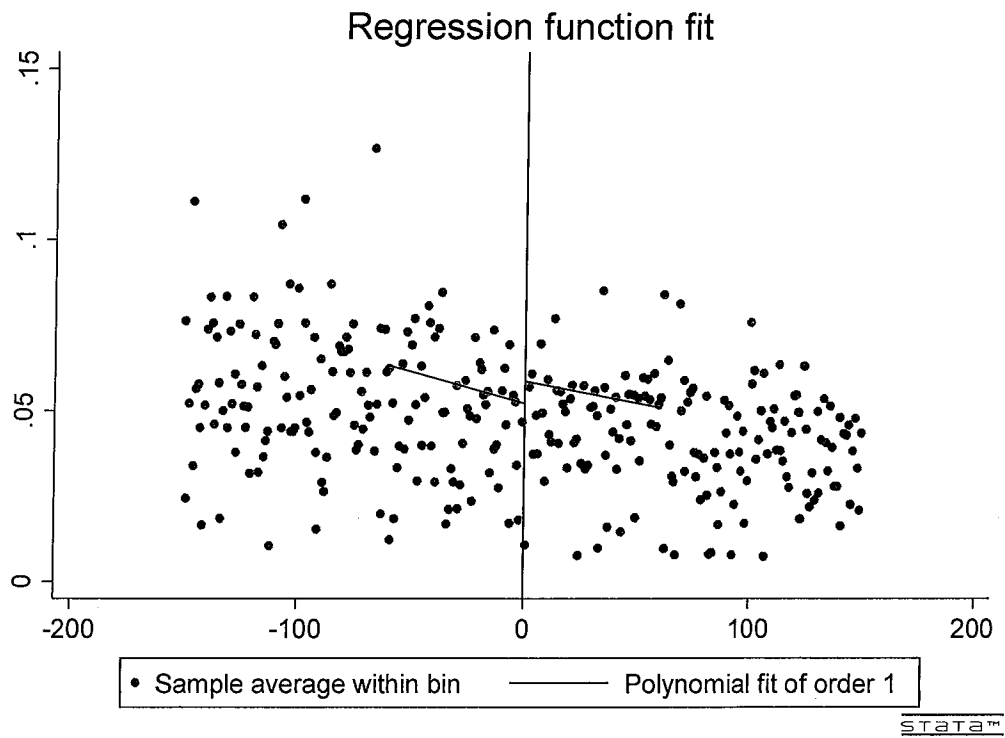
In conclusion however, I would say that there is not very robust evidence for large effects.

ii) I'm most inclined to believe the results when dropping the obs. at 1500gr, using triangular kernel. Further, since we have so many obs., I would tend to use a smaller bandwidth, since that seems to minimize the MSE (by limiting the bias, variance is relatively low, anyway)

7.1) d) 150gr bandwidth, log-bins (other graphs "similar", see do-file)



7.1) 1) Main Specification: $Bw=60$, drop 1500gr, triangular, cluster bweight



```

1  use "/Users/stefan/Desktop/FS19/Econometrics II -
   PhD/PS7/runandjump_sample1500g.dta"
2
3  *a)
4  gen bwtcent=bweight-1500
5
6  *b)
7  gen bin1=bwtcent+0.5
8  hist bin1, width(1) start(-150) freq
9
10 gen modulo10=mod(bwtcent,10)
11 gen bin2=bwtcent+(5-modulo10)
12 hist bin2, width(10) start(-150) freq
13
14 gen modulo25=mod(bwtcent,25)
15 gen bin3=bwtcent+(12.5-modulo25)
16 hist bin3, width(25) start(-150) freq
17
18 *d)
19 *****
20 *****150 bandwidth*****
21 *****
22 ***150 bandwidth, 1gr bin
23 preserve
24 gen frequency=1
25 collapse (count) frequency, by(bin1)
26 *interact bweight with above cutoff
27 gen bin1_interact=0
28 replace bin1_interact=bin1 if bin1<0
29 gen treatment=0
30 replace treatment=1 if bin1<0
31 drop if bin1<-150
32 drop if bin1>150
33 ***regression***
34 reg frequency bin1 bin1_interact treatment, robust
35 predict hat
36 twoway (line hat bin1 if bin1<0) (line hat bin1 if bin1>=0) (bar
   frequency bin1, xline(0, lwidth(vthin)))
37 restore
38
39 ***150 bandwidth, 10gr bin
40 preserve
41 gen frequency=1
42 collapse (count) frequency, by(bin2)
43 *interact bweight with above cutoff
44 gen bin2_interact=0
45 replace bin2_interact=bin2 if bin2<0
46 gen treatment=0
47 replace treatment=1 if bin2<0
48 drop if bin2<-150
49 drop if bin2>150
50 ***regression

```



```

51  reg frequency bin2 bin2_interact treatment, robust
52  predict hat
53  twoway (line hat bin2 if bin2<0) (line hat bin2 if bin2>=0) (bar
54  frequency bin2, xline(0, lwidth(vthin)))
55  restore
56  ***150 bandwidth, 25gr bin
57  preserve
58  gen frequency=1
59  collapse (count) frequency, by(bin3)
60  *interact bweight with above cutoff
61  gen bin3_interact=0
62  replace bin3_interact=bin3 if bin3<0
63  gen treatment=0
64  replace treatment=1 if bin3<0
65  drop if bin3<-150
66  drop if bin3>150
67  ***regression
68  reg frequency bin3 bin3_interact treatment, robust
69  predict hat
70  twoway (line hat bin3 if bin3<0) (line hat bin3 if bin3>=0) (bar
71  frequency bin3, xline(0, lwidth(vthin)))
72  restore
73  *****
74  *****100 bandwidth*****
75  *****
76  ***100 bandwidth, 1gr bin
77  preserve
78  gen frequency=1
79  collapse (count) frequency, by(bin1)
80  *interact bweight with above cutoff
81  gen bin1_interact=0
82  replace bin1_interact=bin1 if bin1<0
83  gen treatment=0
84  replace treatment=1 if bin1<0
85  drop if bin1<-100
86  drop if bin1>100
87  ***regression***
88  reg frequency bin1 bin1_interact treatment, robust
89  predict hat
90  twoway (line hat bin1 if bin1<0) (line hat bin1 if bin1>=0) (bar
91  frequency bin1, xline(0, lwidth(vthin)))
92  restore
93  ***100 bandwidth, 10gr bin
94  preserve
95  gen frequency=1
96  collapse (count) frequency, by(bin2)
97  *interact bweight with above cutoff
98  gen bin2_interact=0
99  replace bin2_interact=bin2 if bin2<0

```



```

100  gen treatment=0
101  replace treatment=1 if bin2<0
102  drop if bin2<-100
103  drop if bin2>100
104  ***regression***
105  reg frequency bin2 bin2_interact treatment, robust
106  predict hat
107  twoway (line hat bin2 if bin2<0) (line hat bin2 if bin2>=0) (bar
frequency bin2, xline(0, lwidth(vthin)))
108  restore
109
110  ***100 bandwidth, 25gr bin
111  preserve
112  gen frequency=1
113  collapse (count) frequency, by(bin3)
114  *interact bweight with above cutoff
115  gen bin3_interact=0
116  replace bin3_interact=bin3 if bin3<0
117  gen treatment=0
118  replace treatment=1 if bin3<0
119  drop if bin3<-100
120  drop if bin3>100
121  ***regression
122  reg frequency bin3 bin3_interact treatment, robust
123  predict hat
124  twoway (line hat bin3 if bin3<0) (line hat bin3 if bin3>=0) (bar
frequency bin3, xline(0, lwidth(vthin)))
125  restore
126
127  *****
128  *****50 bandwidth*****
129  *****
130  ***50 bandwidth, 1gr bin
131  preserve
132  gen frequency=1
133  collapse (count) frequency, by(bin1)
134  *interact bweight with above cutoff
135  gen bin1_interact=0
136  replace bin1_interact=bin1 if bin1<0
137  gen treatment=0
138  replace treatment=1 if bin1<0
139  drop if bin1<-50
140  drop if bin1>50
141  ***regression***
142  reg frequency bin1 bin1_interact treatment, robust
143  predict hat
144  twoway (line hat bin1 if bin1<0) (line hat bin1 if bin1>=0) (bar
frequency bin1, xline(0, lwidth(vthin)))
145  restore
146
147  ***50 bandwidth, 10gr bin
148  preserve

```

```

149 gen frequency=1
150 collapse (count) frequency, by(bin2)
151 *interact bweight with above cutoff
152 gen bin2_interact=0
153 replace bin2_interact=bin2 if bin2<0
154 gen treatment=0
155 replace treatment=1 if bin2<0
156 drop if bin2<-50
157 drop if bin2>50
158 ***regression***
159 reg frequency bin2 bin2_interact treatment, robust
160 predict hat
161 twoway (line hat bin2 if bin2<0) (line hat bin2 if bin2>=0) (bar
frequency bin2, xline(0, lwidth(vthin)))
162 restore
163
164 ***50 bandwidth, 25gr bin
165 preserve
166 gen frequency=1
167 collapse (count) frequency, by(bin3)
168 *interact bweight with above cutoff
169 gen bin3_interact=0
170 replace bin3_interact=bin3 if bin3<0
171 gen treatment=0
172 replace treatment=1 if bin3<0
173 drop if bin3<-50
174 drop if bin3>50
175 ***regression
176 reg frequency bin3 bin3_interact treatment, robust
177 predict hat
178 twoway (line hat bin3 if bin3<0) (line hat bin3 if bin3>=0) (bar
frequency bin3, xline(0, lwidth(vthin)))
179 restore
180
181
182 *e(discontinuity in covariates?)
183 *****
184 ***Mom white***
185 ***Mom educated***
186 ***i, ii, iii***
187 *****
188
189 ssc install rd, replace
190
191 gen mom_white=0
192 replace mom_white=1 if mom_race==1
193 gen treatment=bwtcent<0
194
195 rd mom_white bwtcent, z0(0) bwidth(90) kernel(rectangle) cluster(
bwtcent)
196 rd mom_white bwtcent, z0(0) bwidth(60) kernel(rectangle) cluster(
bwtcent)

```

```

197   rd mom_white bwtcent, z0(0) bwidth(30) kernel(rectangle) cluster(
      bwtcent)
198
199   rd mom_ed1 bwtcent, z0(0) bwidth(90) kernel(rectangle) cluster(
      bwtcent)
200   rd mom_ed1 bwtcent, z0(0) bwidth(60) kernel(rectangle) cluster(
      bwtcent)
201   rd mom_ed1 bwtcent, z0(0) bwidth(30) kernel(rectangle) cluster(
      bwtcent)
202
203
204   ***iv, v, vi*****
205   ***triangular kernel***
206   ***default)*****
207
208   rd mom_white bwtcent, z0(0) bwidth(90) cluster(bwtcent)
209   rd mom_white bwtcent, z0(0) bwidth(60) cluster(bwtcent)
210   rd mom_white bwtcent, z0(0) bwidth(30) cluster(bwtcent)
211
212   rd mom_ed1 bwtcent, z0(0) bwidth(90) cluster(bwtcent)
213   rd mom_ed1 bwtcent, z0(0) bwidth(60) cluster(bwtcent)
214   rd mom_ed1 bwtcent, z0(0) bwidth(30) cluster(bwtcent)
215
216
217   *f)
218   gen ounce_multiple=0
219   replace ounce_multiple=51 if floor(51*28.3495231)==bweight
220   replace ounce_multiple=52 if floor(52*28.3495231)==bweight
221   replace ounce_multiple=53 if floor(53*28.3495231)==bweight
222   replace ounce_multiple=54 if floor(54*28.3495231)==bweight
223
224   replace ounce_multiple=51 if ceil(51*28.3495231)==bweight
225   replace ounce_multiple=52 if ceil(52*28.3495231)==bweight
226   replace ounce_multiple=53 if ceil(53*28.3495231)==bweight
227   replace ounce_multiple=54 if ceil(54*28.3495231)==bweight
228
229   gen ounce51_dummy=ounce_multiple==51
230   gen ounce52_dummy=ounce_multiple==52
231   gen ounce53_dummy=ounce_multiple==53
232   gen ounce54_dummy=ounce_multiple==54
233
234   ***look at jump off trend line at 51, 52, 53, 54 ounces, and
      1500gr
235   ***each with bandwidth of 25 and 100, respectively, for
      mom_white and mom_ed1
236
237   *****
238   ***mom_white***
239   *****
240
241   ***Bandwidth 25
242   preserve

```

```

243
244 drop if bweight<51*28.3495231-25
245 drop if bweight>51*28.3495231+25
246 *generate ounce51_interact=0
247 *replace ounce51_interact=bwtcent if bweight>51*28.3495231
248 reg mom_white ounce51_dummy bwtcent
249
250 restore
251 preserve
252
253 drop if bweight<52*28.3495231-25
254 drop if bweight>52*28.3495231+25
255 reg mom_white ounce52_dummy bwtcent
256
257 restore
258 preserve
259
260 drop if bweight<53*28.3495231-25
261 drop if bweight>53*28.3495231+25
262 drop if bweight==1500
263 reg mom_white ounce53_dummy bwtcent
264
265 restore
266 preserve
267
268 drop if bweight<54*28.3495231-25
269 drop if bweight>54*28.3495231+25
270 reg mom_white ounce54_dummy bwtcent
271
272 restore
273 preserve
274
275 drop if bweight<1500-25
276 drop if bweight<1500+25
277 drop if ounce53_dummy==1
278 drop if ounce52_dummy==1
279 gen gr1500_dummy=bweight==1500
280 reg mom_white gr1500_dummy bwtcent
281
282 restore
283
284 ***Bandwidth 100
285 ***dropping other ounce heaps, and 1500gr observations
286 preserve
287 drop if bweight==1500|ounce52_dummy==1|ounce53_dummy==1|
   ounce54_dummy==1
288 drop if bweight<51*28.3495231-100
289 drop if bweight>51*28.3495231+100
290
291 reg mom_white ounce51_dummy bwtcent
292 restore
293

```

```

294 preserve
295 drop if bweight==1500|ounce51_dummy==1|ounce53_dummy==1|
   ounce54_dummy==1
296 drop if bweight<52*28.3495231-100
297 drop if bweight>52*28.3495231+100
298
299 reg mom_white ounce52_dummy bwtcent
300 restore
301
302 preserve
303 drop if bweight==1500|ounce52_dummy==1|ounce51_dummy==1|
   ounce54_dummy==1
304 drop if bweight<53*28.3495231-100
305 drop if bweight>53*28.3495231+100
306
307 reg mom_white ounce53_dummy bwtcent
308 restore
309
310 preserve
311 drop if bweight==1500|ounce52_dummy==1|ounce51_dummy==1|
   ounce53_dummy==1
312 drop if bweight<54*28.3495231-100
313 drop if bweight>54*28.3495231+100
314
315 reg mom_white ounce54_dummy bwtcent
316 restore
317
318 preserve
319 gen gr1500_dummy=bweight==1500
320 drop if ounce54_dummy==1|ounce52_dummy==1|ounce51_dummy==1|
   ounce53_dummy==1
321 drop if bweight<1500-100|bweight>1500+100
322
323 reg mom_white gr1500_dummy bwtcent
324 restore
325
326
327 *****
328 ***mom_ed1***
329 *****
330
331 ***Bandwidth 25
332 preserve
333
334 drop if bweight<51*28.3495231-25
335 drop if bweight>51*28.3495231+25
336 *generate ounce51_interact=0
337 *replace ounce51_interact=bwtcent if bweight>51*28.3495231
338 reg mom_ed1 ounce51_dummy bwtcent
339
340 restore
341 preserve

```

```

342
343 drop if bweight<52*28.3495231-25
344 drop if bweight>52*28.3495231+25
345 reg mom_ed1 ounce52_dummy bwtcent
346
347 restore
348 preserve
349
350 drop if bweight<53*28.3495231-25
351 drop if bweight>53*28.3495231+25
352 drop if bweight==1500
353 reg mom_ed1 ounce53_dummy bwtcent
354
355 restore
356 preserve
357
358 drop if bweight<54*28.3495231-25
359 drop if bweight>54*28.3495231+25
360 reg mom_ed1 ounce54_dummy bwtcent
361
362 restore
363 preserve
364
365 drop if bweight<1500-25
366 drop if bweight<1500+25
367 drop if ounce53_dummy==1
368 drop if ounce52_dummy==1
369 gen gr1500_dummy=bweight==1500
370 reg mom_ed1 gr1500_dummy bwtcent
371
372 restore
373
374 ***Bandwidth 100
375 ***dropping other ounce heaps, and 1500gr observations
376
377 preserve
378 drop if bweight==1500|ounce52_dummy==1|ounce53_dummy==1|
   ounce54_dummy==1
379 drop if bweight<51*28.3495231-100
380 drop if bweight>51*28.3495231+100
381
382 reg mom_ed1 ounce51_dummy bwtcent
383 restore
384
385 preserve
386 drop if bweight==1500|ounce51_dummy==1|ounce53_dummy==1|
   ounce54_dummy==1
387 drop if bweight<52*28.3495231-100
388 drop if bweight>52*28.3495231+100
389
390 reg mom_ed1 ounce52_dummy bwtcent
391 restore

```

```

392
393   preserve
394   drop if bweight==1500|ounce52_dummy==1|ounce51_dummy==1|
      ounce54_dummy==1
395   drop if bweight<53*28.3495231-100
396   drop if bweight>53*28.3495231+100
397
398   reg mom_ed1 ounce53_dummy bwtcent
399   restore
400
401   preserve
402   drop if bweight==1500|ounce52_dummy==1|ounce51_dummy==1|
      ounce53_dummy==1
403   drop if bweight<54*28.3495231-100
404   drop if bweight>54*28.3495231+100
405
406   reg mom_ed1 ounce54_dummy bwtcent
407   restore
408
409   preserve
410   gen gr1500_dummy=bweight==1500
411   drop if ounce54_dummy==1|ounce52_dummy==1|ounce51_dummy==1|
      ounce53_dummy==1
412   drop if bweight<1500-100|bweight>1500+100
413
414   reg mom_ed1 gr1500_dummy bwtcent
415   restore
416
417
418   *g)
419   *****Robust*****
420   ***same as in e)***
421   ***rect. kernel***
422   rd agedth5 bwtcent, z0(0) bwidth(90) kernel(rectangle) robust
423   rd agedth5 bwtcent, z0(0) bwidth(60) kernel(rectangle) robust
424   rd agedth5 bwtcent, z0(0) bwidth(30) kernel(rectangle) robust
425
426   ***triangular kernel
427   *****default)*****
428   rd agedth5 bwtcent, z0(0) bwidth(90) robust
429   rd agedth5 bwtcent, z0(0) bwidth(60) robust
430   rd agedth5 bwtcent, z0(0) bwidth(30) robust
431
432
433   *****Cluster*****
434   ***same as in e)***
435   ***rect. kernel***
436   rd agedth5 bwtcent, z0(0) bwidth(90) kernel(rectangle) robust
      cluster(bwtcent)
437   rd agedth5 bwtcent, z0(0) bwidth(60) kernel(rectangle) robust
      cluster(bwtcent)
438   rd agedth5 bwtcent, z0(0) bwidth(30) kernel(rectangle) robust

```



```

cluster(bwtcent)
439
440 ***triangular kernel
441 *****default*****
442 rd agedth5 bwtcent, z0(0) bwidth(90) robust cluster(bwtcent)
443 rd agedth5 bwtcent, z0(0) bwidth(60) robust cluster(bwtcent)
444 rd agedth5 bwtcent, z0(0) bwidth(30) robust cluster(bwtcent)
445
446
447 *h)
448 ***drop 1500gr obs.
449 ***repeat exercise g) (same specs.)
450 preserve
451 drop if bweight==1500
452
453 ***Robust***
454 *same as in e)
455 *rect. kernel
456 rd agedth5 bwtcent, z0(0) bwidth(90) kernel(rectangle) robust
457 rd agedth5 bwtcent, z0(0) bwidth(60) kernel(rectangle) robust
458 rd agedth5 bwtcent, z0(0) bwidth(30) kernel(rectangle) robust
459
460 *triangular kernel
461 rd agedth5 bwtcent, z0(0) bwidth(90) robust
462 rd agedth5 bwtcent, z0(0) bwidth(60) robust
463 rd agedth5 bwtcent, z0(0) bwidth(30) robust
464
465 ***Cluster
466 *same as in e)
467 *rect. kernel
468 rd agedth5 bwtcent, z0(0) bwidth(90) kernel(rectangle) robust
469 cluster(bwtcent)
470 rd agedth5 bwtcent, z0(0) bwidth(60) kernel(rectangle) robust
471 cluster(bwtcent)
472 rd agedth5 bwtcent, z0(0) bwidth(30) kernel(rectangle) robust
473 cluster(bwtcent)
474
475 ***triangular kernel
476 *****default*****
477 rd agedth5 bwtcent, z0(0) bwidth(90) robust cluster(bwtcent)
478 rd agedth5 bwtcent, z0(0) bwidth(60) robust cluster(bwtcent)
479 rd agedth5 bwtcent, z0(0) bwidth(30) robust cluster(bwtcent)
480
481 restore
482
483 *****drop obs. measured in ounces
484 ***repeat exercise g) (same specs.)
485 preserve
486 drop if ounce==1
487
488 ***Robust***
489 *same as in e)

```

```

487 *rect. kernel
488 rd agedth5 bwtcent, z0(0) bwidth(90) kernel(rectangle) robust
489 rd agedth5 bwtcent, z0(0) bwidth(60) kernel(rectangle) robust
490 rd agedth5 bwtcent, z0(0) bwidth(30) kernel(rectangle) robust
491
492 ***triangular kernel
493 rd agedth5 bwtcent, z0(0) bwidth(90) robust
494 rd agedth5 bwtcent, z0(0) bwidth(60) robust
495 rd agedth5 bwtcent, z0(0) bwidth(30) robust
496
497 ***Cluster
498 *same as in e)
499 *rect. kernel
500 rd agedth5 bwtcent, z0(0) bwidth(90) kernel(rectangle) cluster(
bwtcent)
501 rd agedth5 bwtcent, z0(0) bwidth(60) kernel(rectangle) cluster(
bwtcent)
502 rd agedth5 bwtcent, z0(0) bwidth(30) kernel(rectangle) cluster(
bwtcent)
503
504 ***triangular kernel
505 *****default)*****
506 rd agedth5 bwtcent, z0(0) bwidth(90) cluster(bwtcent)
507 rd agedth5 bwtcent, z0(0) bwidth(60) cluster(bwtcent)
508 rd agedth5 bwtcent, z0(0) bwidth(30) cluster(bwtcent)
509
510 *when clustering on bweight, SEs considerably larger. indicates
that we should
511 *cluster on bweight
512
513 restore
514
515
516 *i)
517 *in light of exercise h), I would prefer a specification with
clustering on
518 *bweight. Further, I would use the triangular kernel because
whilst considering
519 *possibly more data points, the fact that it down-weights obs.
far away means
520 *that the bias tend to be smaller.
521 *Next, I'll use the suggested bandwidth of 60 and compare the
results of RD
522 *with triangular kernel, cluster (bwtcent)
523 *for both keeping/dropping the obs. of 1500gr
524
525 ssc install rdrobust
526 preserve
527 drop if bweight==1500
528 rd agedth5 bwtcent, z0(0) bwidth(60) cluster(bwtcent)
529 rdplot agedth5 bwtcent, p(1) z0(0) h(60) cluster(bwtcent)
530 restore

```

Question 2

- a) Cultural background is the treatment, i.e., whether the mother is Swedish or of foreign origin (in the 2nd step, could also differentiate between different countries of origin of the immigrants).

The outcome is birth-spacing, or more generally fertility. In a broader sense, also economic outcomes such as school performance of the child could be considered an outcome.

- b) Unmeasured immigrant characteristics.

The immigrants are a non-representative subgroup of their countries of origin, since there was some sort of self-selection going on, which resulted in migrating to Sweden.

For example, I would expect that immigrants tend to be more liberal, ambitious, educated than the "average", representative resident in their country of origin.

This is a problem for identification, in particular for the epidemiological approach to culture, and also for the DiD. It affects moreover the external validity. Also, prior to the policy change, Swedish natives and immigrants possibly faced different incentives (possibly institutional differences).

- c) Their "DiD" approach likely assumes that after the policy change, both natives / immigrants face the same circumstances and have access to the same level of "opportunities".

The crucial assumption is that any change, other than the policy change, that influences the outcome, affects both groups the same / equally. This is the common / parallel trend assumption.

d)

Although the parallel trends assumption cannot be tested, we can compare the pre-policy periods for internal validity.

→ Parallel trends, both flat and parallel for both groups, thus internally validity satisfied.

External validity: cannot tell whether or not satisfied.

e)

I think the general problem here is about external validity.

Some immigrants might self-select into migrating to Sweden.

This could make them fundamentally different.

Furthermore, and more generally, it might be that differences in birth-spacing between the two groups prior to the changes affect how "responsive" the birth-spacing-durations are to the fertility policy. Could divide the sample into immigrants closer to Sweden as country of origin vs. countries further away (Arabian Peninsula), and redo the DiD for Swedish vs. "close proximity to Sweden" immigrants.

Additionally, another concern for inference could also be that the composition of migrants over time changes, possibly induced by policy).