

```

1 #delimit ;
2
3 capture clear all ;
4 capture log close ;
5
6 log using "...\\program1--prepare data.log", replace ;
7
8 *****
9 * Author: E McClintock *
10 * Date last edited: Aug, 2014 *
11 * Last edits: Added comments for *
12 * posting online and deleted *
13 * tables with small cell sizes *
14 * Does: preps data for exchange & *
15 * matching analysis, uses "ice" *
16 * for multiple imputation *
17 *****;
18
19 *** Open Partner-Level Data ***;
20
21 /* NOTES:
22
23 This data is organized by gender. After cleaning & prepping variables I link respondents
to their recruited partners. By "respondent" I mean original Add Health respondent and by
"partner" I mean the recruited partner. I organize these couples by gender. This is
reflected in the naming conventions. A variable that begins with f#_ refers to information
about the female partner in Wave #. A variable that begins with m#_ refers to information
about the male partner in Wave #. For example, f3_yrsedu is the female partner's years of
completed education at the Wave III (3) interview. If there is no # (f_ or m_) the
variable is not time-variant. For example, f_white means that the female partner is
white--race does not change (much) over time. Obviously, recruited partners are only
interviewed once, in Wave III. However, the partners complete a slightly-modified version
of the Wave III interview which asks some retrospective questions which I use to
approximate information collected on main respondent in earlier waves (e.g., father's
occupational status when the partner was an adolescent). Therefore it is possible for
partners to have values for some variables that are tagged as Wave I (1). Also, in program
2 I create forecast measures so it is possible for partners to have values on f4_...p and
m4_...p variables (p for predicted). In the program following this program I use
information on respondent's income/SEI/etc at Wave IV to forecast WAVE IV income/SEI/etc
for respondents and partners. This provides an approximation of expected future income at
Wave III when individuals would be gauging their partner's economic potential. A variable
that begins with c#_ is about the couple. For example, c_white means that the couple is
white (both white non-Hispanic).
24
25 I am not posting the programs in which I clean and prep basic variables. These programs
are simple and it is not worth my time to edit them (write comments, delete any tables with
small sample sizes from log files, etc.) when they could be easily reproduced by anyone
with access to the Add Health data and codebooks.
26
27 */
28
29 use "...\\partners.dta", clear ;
30
31 des, short ;
32
33 *** Construct ***;
34
35 ** female partner is currently pregnant **;
36
37 * Note: I have the same N for this as Carmalt et al 2008--which I should--this is simply a
verification that we have both prepared the data in the same way *;
38 tab f3_pregnow ;
39
40 ** partner is physically attractive **;
41
42 * Note: I have the same N for this as Carmalt et al 2008--which I should--this is simply a
verification that we have both prepared the data in the same way *;
43 recode m3_physatt 1/3=0 4/5=1, gen(m3_attvatt) ;
44 recode f3_physatt 1/3=0 4/5=1, gen(f3_attvatt) ;
45

```

```

46 sum m3_attvatt f3_attvatt ;
47 sum m3_attvatt f3_attvatt if f3_pregnow == 0 ;
48
49 * body mass index *;
50
51 * Note: Carmalt et al 2008 report more obs than I do - I am not sure how that could be.
However, the difference is minor and I do not use BMI in my analysis. There are a lot of
reasonable ways to calculate BMI which yield slightly different Ns (hence my various
version below). For one thing, some people max the scale out at 330 lbs. Using 330 would
yield a lower BMI estimate than would using their true (unknown but heavier than 330)
weight. The researcher's decision about that alters the N. For another, in some cases the
respondent has information on (for example) complete inches but is missing data on partial
inches. Using inches alone when partial inches are missing would yield a
reasonably-accurate estimate of height. Or a researcher might count that respondent as
missing information on height. I tried several approaches to calculating BMI.
Fortunately, for most respondents it really does not make a difference. ;
52 sum m3_bmi m3_bmi_1 m3_bmi_2 m3_bmi_3 f3_bmi f3_bmi_1 f3_bmi_2 f3_bmi_3 ;
53 sum m3_bmi m3_bmi_1 m3_bmi_2 m3_bmi_3 f3_bmi f3_bmi_1 f3_bmi_2 f3_bmi_3 if f3_pregnow == 0 ;
54
55 * weight cats *;
56
57 sum m3_obese m3_overweight m3_normalweight m3_underweight f3_obese f3_overweight
f3_normalweight f3_underweight ;
58 sum m3_obese m3_overweight m3_normalweight m3_underweight f3_obese f3_overweight
f3_normalweight f3_underweight if f3_pregnow == 0 ;
59
60 ** physical attractiveness **;
61
62 * Note: I have the same N for this as Carmalt et al 2008--which I should--this is simply a
verification that we have both prepared the data in the same way *;
63 sum m3_physatt f3_physatt ;
64 sum m3_physatt f3_physatt if f3_pregnow == 0 ;
65
66 ** groomed **;
67
68 * Note: I have the same N for this as Carmalt et al 2008--which I should--this is simply a
verification that we have both prepared the data in the same way *;
69 sum m3_groomed f3_groomed ;
70 sum m3_groomed f3_groomed if f3_pregnow == 0 ;
71
72 recode m3_groomed 1/3=0 4/5=1, gen(m3_wellg) ;
73 recode f3_groomed 1/3=0 4/5=1, gen(f3_wellg) ;
74 sum m3_wellg f3_wellg if f3_pregnow == 0 ;
75
76 ** years of edu **;
77
78 * Note: I have the same N for this as Carmalt et al 2008--which I should--this is simply a
verification that we have both prepared the data in the same way *;
79 sum m3_yrsedu f3_yrsedu ;
80 sum m3_yrsedu f3_yrsedu if f3_pregnow == 0 ;
81
82 ** log of personal income **;
83
84 * Note: Carmalt et al 2008 have more observations for ln income than I do - I think they
use the "best guess" measure to fill in missing values. I'm sure I'm using the same basic
variable they are - sect 15, question h3ec2. I have checked my work and I think I have the
correct # of valid obs for this (without using other vars to fill in missing info). ;
85 sum m3_ln_income f3_ln_income ;
86 sum m3_ln_income f3_ln_income if f3_pregnow == 0 ;
87 sum m3_income f3_income if f3_pregnow == 0 ;
88
89 ** AH_PVT **;
90
91 * Note: I do NOT have the same N for this as Carmalt et al 2008 - I have a larger N.
Again, I checked my work and do not see any sources of error. *;
92 sum m3_ahpvt f3_ahpvt ;
93 sum m3_ahpvt f3_ahpvt if f3_pregnow == 0 ;
94
95 ** personality attractiveness **;
96

```

```

97 * Note: I have the same N for this as Carmalt et al 2008--which I should--this is simply a
98 verification that we have both prepared the data in the same way *;
99 sum m3_peratt f3_peratt ;
100 sum m3_peratt f3_peratt if f3_pregnow == 0 ;
101 recode m3_peratt 1/3=0 4/5=1, gen(m3_attp) ;
102 recode f3_peratt 1/3=0 4/5=1, gen(f3_attp) ;
103 sum m3_attp f3_attp if f3_pregnow == 0 ;
104
105 ** emotional supportiveness **;
106
107 sum m3_rmood m3_laugh m3_frsta f3_rmood f3_laugh f3_frsta ;
108
109 gen m3_emosup = m3_rmood + m3_laugh + m3_frsta ;
110 gen f3_emosup = f3_rmood + f3_laugh + f3_frsta ;
111
112 * Note: I do NOT have the same N for this as Carmalt et al 2008 - I have a smaller N. The
reason that I have a smaller N is because they ignore missing data. They sum rmood, laugh,
and frsta to create an index with values from 0 to 12. But when data are missing they
still sum it even though this would give an inaccurately low emosup score (0 to 8 or 0 to
4, depending if 1 or 2 variables are missing). Their Ns (1304 for females and 1293 for
females) are the Ns for non-missing on at least one of the 3 variables. My Ns are
non-missing on all 3. So neither of us made an error in constructing this but we differ in
how we choose to deal with missing data. ;
113 sum m3_emosup f3_emosup ;
114 sum m3_emosup f3_emosup if f3_pregnow == 0 ;
115
116 ** age **;
117
118 * Note: I do have the same N for this as Carmalt et al 2008 - but we have very slightly
different means. The variable calcage3 is a prepared variable provided by Add Health.
Perhaps Carmalt et al 2008 used information on birthdate and interview date to estimate
age? *;
119 sum m3_calcage3 f3_calcage3 ;
120 sum m3_calcage3 f3_calcage3 if f3_pregnow == 0 ;
121
122 ** race groups **;
123
124 * Note: I do NOT have the same N for this as Carmalt et al 2008 - I have a larger N because
I use interviewer's report to fill in missing values. Again, neither of us did anything
wrong. We just deal with missing values differently. *;
125 sum m3_white m3_black m3_otall m3_hisp f3_white f3_black f3_otall f3_hisp ;
126 sum m3_white m3_black m3_otall m3_hisp f3_white f3_black f3_otall f3_hisp if f3_pregnow == 0
;
127
128 * relationship duration *;
129
130 * Note: I do NOT have the same N for this as Carmalt et al 2008 - I have a much smaller N -
but don't see how they could have done any different assuming that we are using the same
underlying measure. *;
131 sum m3_rdur f3_rdur ;
132 sum m3_rdur f3_rdur if f3_pregnow == 0 ;
133 replace m3_rdur = . if m3_rdur < 0 ;
134 replace f3_rdur = . if f3_rdur < 0 ;
135 sum m3_rdur f3_rdur ;
136 sum m3_rdur f3_rdur if f3_pregnow == 0 ;
137
138 * marital status *;
139
140 * Note: I do have ~the same N for this as Carmalt et al 2008 *;
141 sum c3_marital c3_dating c3_cohab c3_married f3_nowmrd f3_nowchb m3_nowmrd m3_nowchb ;
142 sum c3_marital c3_dating c3_cohab c3_married f3_nowmrd f3_nowchb m3_nowmrd m3_nowchb if
f3_pregnow == 0 ;
143
144 gen f3_marital = f3_nowmrd ;
145 * marriage trumps cohabitation ;
146 replace f3_marital = 2 if f3_nowchb == 1 & f3_marital ~= 1 ;
147 * if missing, use partner's information ;
148 replace f3_marital = 1 if m3_nowmrd == 1 & (f3_marital == . | f3_marital == 0);
149 replace f3_marital = 2 if m3_nowchb == 1 & (f3_marital == . | f3_marital == 0);

```

```

150
151 gen m3_marital = m3_nowmrd ;
152 * marriage trumps cohabitation ;
153 replace m3_marital = 2 if m3_nowchb == 1 & m3_marital ~= 1 ;
154 * if missing, use partner's information ;
155 replace m3_marital = 1 if f3_nowmrd == 1 & (m3_marital == . | m3_marital == 0);
156 replace m3_marital = 2 if f3_nowchb == 1 & (m3_marital == . | m3_marital == 0);
157
158 * for the most part, they agree--as they ought to ;
159 tab1 f3_marital m3_marital ;
160 tab1 f3_marital m3_marital if f3_pregnow == 0 ;
161
162 drop f3_now* ;
163 * if not married or cohabiting then dating ;
164 replace f3_marital = 3 if f3_marital == 0 ;
165
166 recode f3_marital      1=1 2/3=0, gen(f3_married) ;
167 recode f3_marital      1=0 2=1 3=0,   gen(f3_cohab) ;
168 recode f3_marital      1/2=0 3=1,     gen(f3_dating) ;
169 tab f3_marital ; sum f3_married f3_cohab f3_dating ;
170
171 drop m3_now* ;
172 * if not married or cohabiting then dating ;
173 replace m3_marital = 3 if m3_marital == 0 ;
174
175 *tab f3_marital m3_marital, miss ;
176 replace m3_marital=f3_marital if m3_marital==. ;
177
178 recode m3_marital      1=1 2/3=0, gen(m3_married) ;
179 recode m3_marital      1=0 2=1 3=0,   gen(m3_cohab) ;
180 recode m3_marital      1/2=0 3=1,     gen(m3_dating) ;
181 tab m3_marital ; sum m3_married m3_cohab m3_dating ;
182
183 label values f3_marital marital ;
184 label values m3_marital marital ;
185
186 tab1 m3_marital f3_marital ;
187
188 sum m3_married m3_cohab m3_dating f3_married f3_cohab f3_dating ;
189 sum m3_married m3_cohab m3_dating f3_married f3_cohab f3_dating if f3_pregnow == 0 ;
190
191 * check race variables and make categorical vars ;
192
193 sum m3_white m3_black m3_other m3_asian m3_amind m3_hisp ;
194 gen checkm=m3_white+m3_black+m3_other+m3_asian+m3_amind ;
195 tab checkm ;
196 *tab checkm m3_hisp, miss ;
197 drop checkm ;
198 gen m3_race3 = 1 if m3_white==1 ;
199 replace m3_race3 = 2 if m3_black==1 ;
200 replace m3_race3 = 3 if m3_hisp==1 | m3_other==1 | m3_amind==1 | m3_asian==1 ;
201 tab m3_race3, miss ;
202 gen m3_race4 = 1 if m3_white==1 ;
203 replace m3_race4 = 2 if m3_black==1 ;
204 replace m3_race4 = 3 if m3_hisp==1 ;
205 replace m3_race4 = 4 if m3_other==1 | m3_amind==1 | m3_asian==1 ;
206 tab m3_race4, miss ;
207 drop m3_other m3_white m3_black m3_asian m3_amind m3_hisp ;
208 recode m3_race4      1=1 2/4=0, gen(m3_white) ;
209 recode m3_race4      1=0 2=1 3/4=0, gen(m3_black) ;
210 recode m3_race4      1/2=0 3=1 4=0, gen(m3_hisp) ;
211 recode m3_race4      1/3=0 4=1      , gen(m3_other) ;
212 tab m3_race4 ;
213 sum m3_white m3_black m3_hisp m3_other ;
214 gen checkm=m3_white+m3_black+m3_hisp+m3_other ;
215 tab checkm ; drop checkm ;
216
217 sum f3_white f3_black f3_other f3_asian f3_amind f3_hisp ;
218 gen checkf=f3_white+f3_black+f3_other+f3_asian+f3_amind ;
219 *tab checkf ;

```

```

220 *tab checkf f3_hisp, miss ;
221 drop checkf ;
222 gen f3_race3 = 1 if f3_white==1 ;
223 replace f3_race3 = 2 if f3_black==1 ;
224 replace f3_race3 = 3 if f3_hisp==1 | f3_other==1 | f3_amind==1 | f3_asian==1 ;
225 *tab f3_race3, miss ;
226 gen f3_race4 = 1 if f3_white==1 ;
227 replace f3_race4 = 2 if f3_black==1 ;
228 replace f3_race4 = 3 if f3_hisp==1 ;
229 replace f3_race4 = 4 if f3_other==1 | f3_amind==1 | f3_asian==1 ;
230 *tab f3_race4, miss ;
231 drop f3_other f3_white f3_black f3_asian f3_amind f3_hisp ;
232 recode f3_race4 1=1 2/4=0, gen(f3_white) ;
233 recode f3_race4 1=0 2=1 3/4=0, gen(f3_black) ;
234 recode f3_race4 1/2=0 3=1 4=0, gen(f3_hisp) ;
235 recode f3_race4 1/3=0 4=1 , gen(f3_other) ;
236 tab f3_race4 ;
237 sum f3_white f3_black f3_hisp f3_other ;
238 gen checkm=f3_white+f3_black+f3_hisp+f3_other ;
239 tab checkm ; drop checkm ;
240
241 tab m3_race3 f3_race3 ;
242 *tab m3_race4 f3_race4 ;
243
244 * citizenship--a reviewer wanted it *;
245
246 pwcorr m3_citizen f3_citizen, sig ;
247 tab m3_citizen f3_citizen, chi2 ;
248
249 * weight category *;
250
251 * make weight cat var *;
252
253 label define weightcat
254 1 underweight
255 2 normalweight
256 3 overweight
257 4 obese ;
258 foreach p in f3 m3 { ;
259
260 gen `p' weightcat = `p' underweight ;
261 replace `p' weightcat = 2 if `p' normalweight == 1 ;
262 replace `p' weightcat = 3 if `p' overweight == 1 ;
263 replace `p' weightcat = 4 if `p' obese == 1 ;
264 label values `p' weightcat weightcat ;
265 label var `p' weightcat "Weight category" ;
266 tab `p' weightcat ;
267 table `p' weightcat, contents(min `p' bmi mean `p' bmi max `p' bmi) ;
268
269 } ;
270
271 sum
272 f3_bmi m3_bmi f3_bmi_3 m3_bmi_3
273 f3_ln_income m3_ln_income
274 f3_yrsedu m3_yrsedu
275 f3_ee_cgrdp m3_ee_cgrdp
276 f3_ahpvt m3_ahpvt
277 f3_calcage3 m3_calcage3
278 f3_race3 f3_white f3_black f3_other m3_race3 m3_white m3_black m3_other
279 f3_emosup m3_emosup
280 f3_rdur m3_rdur
281 f3_marital f3_married f3_cohab f3_dating m3_marital m3_married m3_cohab m3_dating
282 f3_pregnow
283 m1_dad_hh7 f1_dad_hh7 m1_mom_hh7 f1_mom_hh7 m3_hh7 f3_hh7 f4_hh7 m4_hh7
284 m3_sei f3_sei m4_sei f4_sei f1_dad_sei m1_dad_sei
285 ;
286
287 tab1
288 f3_physatt m3_physatt f3_peratt m3_peratt f3_groomed m3_groomed
289 f3_underweight m3_underweight f3_normalweight m3_normalweight f3_overweight m3_overweight

```

```

290 f3_obese m3_obese ;
291 * The code below allows me to identify which information was imputed for these variables ;
292
293 gen f3_miss_rdur = 0 if f3_rdur ~= . ;
294 replace f3_miss_rdur = 1 if f3_rdur == . ;
295
296 gen m3_miss_rdur = 0 if m3_rdur ~= . ;
297 replace m3_miss_rdur = 1 if m3_rdur == . ;
298
299 gen c3_miss_rdur = 0 if m3_rdur ~= . | f3_rdur ~= . ;
300 replace c3_miss_rdur = 1 if m3_rdur == . & f3_rdur == . ;
301
302 gen f3_vfat = 0 if f3_bmi ~= . ;
303 replace f3_vfat = 1 if f3_bmi ~= . & (f3_bmi > f3_bmi_3) ;
304 sum f3 bmi f3 bmi 3 f3 vfat ;
305
306 gen m3_vfat = 0 if m3_bmi ~= . ;
307 replace m3_vfat = 1 if m3_bmi ~= . & (m3_bmi > m3_bmi_3) ;
308 sum m3_bmi m3_bmi_3 m3_vfat ;
309
310 foreach l in m f { ;
311 foreach p in mom dad { ;
312 recode `l'1_`p'_edu 1/3=0 4=1, gen(`l'1_`p'_cg) ;
313 } ; } ;
314
315 recode f4_edu5 1/3=0 4/5=1, gen(f4_cgp) ;
316 recode m4_edu5 1/3=0 4/5=1, gen(m4_cgp) ;
317
318 ***** Impute data *****;
319
320 keep
321 aid f3_partner
322 c3_sameint *intedu *intid *intrace
323 f3_physatt m3_physatt f3_peratt m3_peratt f3_groomed m3_groomed f3_attract m3_attract
324 f3_bmi_3 m3_bmi_3
325 f3_ln_income m3_ln_income f3_income m3_income
326 f3_yrsedu m3_yrsedu
327 f3_ee_cgrdp m3_ee_cgrdp
328 f3_ahpvt m3_ahpvt
329 f3_calcage3 m3_calcage3
330 f3_race4 f3_white f3_black f3_hisp f3_other m3_race4 m3_white m3_black m3_hisp m3_other
331 f3_emosup m3_emosup
332 f3_rdur m3_rdur
333 f3_marital f3_married f3_cohab f3_dating m3_marital m3_married m3_cohab m3_dating
334 f3_pregnow
335 m1_dad_hh7 f1_dad_hh7 m1_mom_hh7 f1_mom_hh7 m3_hh7 f3_hh7 f4_hh7 m4_hh7
336 m1_dad_edu f1_dad_edu m1_mom_edu f1_mom_edu f1_hshld_inc m1_hshld_inc
337 f2_gpa m2_gpa
338 f3_citizen m3_citizen
339 f3_cesd9 m3_cesd9 f3_diagdep m3_diagdep f3_health m3_health
340 m3_sei m3_npboss90
341 f3_sei f3_npboss90
342 m4_sei m4_npboss90
343 f4_sei f4_npboss90
344 f1_dad_sei f1_dad_npboss90
345 m1_dad_sei m1_dad_npboss90
346 f4_inc m4_inc
347 f4_edu5 m4_edu5
348 f3_inschl3 m3_inschl3
349 f3_trdgdr m3_trdgdr
350 ;
351
352 sum ; des ;
353
354 *** Impute Missing Data ***;
355
356 /***** Canceled out because there is no need to re-run it and it takes forever *****/
357
358 * I only need to re-run it when something has changed. At the moment I am just running the

```

```

program to create a log file to post online after deleting tables with small cell sizes and
adding comments. ;
359
360 ice
361   f3_physatt m3_physatt f3_peratt m3_peratt f3_groomed m3_groomed f3_attract m3_attract
362   f3_calcage3 m3_calcage3
363   f3_bmi_3 m3_bmi_3
364   f3_ln_income m3_ln_income f3_inc m3_inc
365   f3_yrsedu m3_yrsedu
366   f3_ee_cgrdp m3_ee_cgrdp
367   f3_ahpvt m3_ahpvt
368   f3_emosup m3_emosup
369   f3_rdur m3_rdur
370   m3_race4 m3_white m3_black m3_hisp m3_other f3_race4 f3_white f3_black f3_hisp f3_other
371   f3_married f3_cohab
372   m1_dad hh7 f1_dad hh7 m1_mom hh7 f1_mom hh7 m3_hh7 f3_hh7 f4_hh7 m4_hh7
373   m1_dad_edu f1_dad_edu m1_mom_edu f1_mom_edu f1_hshld_inc m1_hshld_inc
374   f2_gpa m2_gpa
375   f4_edu5 m4_edu5 f4_inc m4_inc
376   f3_citizen m3_citizen
377   f3_cesd9 m3_cesd9 f3_diagdep m3_diagdep f3_health m3_health
378   f3_inschl3 m3_inschl3
379   f3_trdgdr m3_trdgdr
380   m3_sei f3_sei m4_sei f4_sei f1_dad_sei m1_dad_sei,
381   saving("...\prepare_data.dta")
382   m(10)
383   seed(1285964)
384   sub(m3_race4: m3_black m3_other, f3_race4: f3_black f3_other)
385   cmd(m1_dad_hh7 f1_dad_hh7 m1_mom_hh7 f1_mom_hh7 m3_hh7 f3_hh7 f4_hh7 m4_hh7 f4_edu5 m4_edu5
m1_dad_edu f1_dad_edu m1_mom_edu f1_mom_edu f3_attract m3_attract f3_health m3_health
f3_trdgdr m3_trdgdr f3_yrsedu m3_yrsedu:ologit, f3_inschl3 m3_inschl3: mlogit)
386   passive(m3_white:(m3_race4==1) \m3_black:(m3_race4==2) \m3_hisp:(m3_race4==3)
\m3_other:(m3_race4==4)
387   \f3_white:(f3_race4==1) \f3_black:(f3_race4==2) \f3_hisp:(f3_race4==3)
\f3_other:(f3_race4==4) )
388   eq(f3_race: m3_black m3_other f3_ee_cgrdp m3_ee_cgrdp f1_dad hh7,
389   m3_race: f3_black f3_other f3_ee_cgrdp m3_ee_cgrdp m1_dad_hh7)
390   replace ;
391
392   ***** Canceled out because there is no need to re-run it and it takes forever *****/
393
394   use "...\prepare_data.dta", clear ;
395
396   des, short ;
397
398   *tab m3_race4 f3_race4 ;
399   tab m3_race4 m3_white ;
400   tab m3_race4 m3_black ;
401   tab m3_race4 m3_hisp ;
402   tab m3_race4 m3_other ;
403   tab f3_race4 f3_white ;
404   tab f3_race4 f3_black ;
405   tab f3_race4 f3_hisp ;
406   tab f3_race4 f3_other ;
407   drop *_white *_black *_hisp *_other ;
408   recode m3_race4 1=1 2/4=0, gen(m3_white) ;
409   recode m3_race4 1=0 2=1 3/4=0, gen(m3_black) ;
410   recode m3_race4 1/2=0 3=1 4=0, gen(m3_hisp) ;
411   recode m3_race4 1/3=0 4=1 , gen(m3_other) ;
412   recode f3_race4 1=1 2/4=0, gen(f3_white) ;
413   recode f3_race4 1=0 2=1 3/4=0, gen(f3_black) ;
414   recode f3_race4 1/2=0 3=1 4=0, gen(f3_hisp) ;
415   recode f3_race4 1/3=0 4=1 , gen(f3_other) ;
416   *tab m3_race4 f3_race4 ;
417   sum *_white *_black *_hisp *_other ;
418
419   recode m4_edu5 1=0 2=1 3/5=0, gen(m4_hsged) ;
420   recode m4_edu5 1/2=0 3=1 4/5=0, gen(m4_mths) ;
421   recode m4_edu5 1/3=0 4/5=1, gen(m4_cgrdp) ;
422

```

```

423 recode f4_educ5 1=0 2=1 3/5=0, gen(f4_hsged) ;
424 recode f4_educ5 1/2=0 3=1 4/5=0, gen(f4_mths) ;
425 recode f4_educ5 1/3=0 4/5=1, gen(f4_cgrdp) ;
426
427 * Create couple race vars *;
428
429 * NOTE: As discussed in the article, I use couple race because most couples are
same-race. His and her race are very collinear. ;
430
431 *tab f3_race4 m3_race4 ;
432 gen c3_white = 0 if m3_race4 ~= . & f3_race4 ~= . ;
433 gen c3_black = 0 if m3_race4 ~= . & f3_race4 ~= . ;
434 gen c3_hisp = 0 if m3_race4 ~= . & f3_race4 ~= . ;
435 gen c3_other = 0 if m3_race4 ~= . & f3_race4 ~= . ;
436 replace c3_white = 1 if m3_white == 1 & f3_white == 1 ;
437 replace c3_black = 1 if m3_black == 1 & f3_black == 1 ;
438 replace c3_hisp = 1 if m3_hisp == 1 & f3_hisp == 1 ;
439 replace c3_other = 1 if m3_other == 1 & f3_other == 1 ;
440 replace c3_race4 = 1 if m3_race4 == f3_race4 & m3_race4 ~= . & f3_race4 ~= . ;
441 gen c3_race4 = 1 if c3_white==1 ;
442 replace c3_race4 = 2 if c3_black==1 ;
443 replace c3_race4 = 3 if c3_hisp==1 ;
444 replace c3_race4 = 4 if c3_other==1 ;
445 label define race4 1 white 2 black 3 hispanic 4 other ;
446 label values c3_race4 race4 ;
447 gen c3_race5 = c3_race4 ;
448 tab c3_race4 ;
449 replace c3_race5 = 5 if m3_race4 ~= f3_race4 & m3_race4 ~= . & f3_race4 ~= . ;
450 label define race5 1 white 2 black 3 hispanic 4 other 5 mixed ;
451 label values c3_race5 race5 ;
452 tab c3_race5 ;
453
454 tab c3_race4 c3_race5 ;
455
456 * Create couple marital status vars *;
457
458 *tab f3_marital m3_marital ;
459 gen c3_married = 0 if m3_marital ~= . & f3_marital ~= . ;
460 gen c3_cohab = 0 if m3_marital ~= . & f3_marital ~= . ;
461 gen c3_dating = 0 if m3_marital ~= . & f3_marital ~= . ;
462 gen c3_mrtlmix = 0 if m3_marital ~= . & f3_marital ~= . ;
463 replace c3_married = 1 if m3_married == 1 & f3_married == 1 ;
464 replace c3_cohab = 1 if m3_cohab == 1 & f3_cohab == 1 ;
465 replace c3_dating = 1 if m3_dating == 1 & f3_dating == 1 ;
466 replace c3_mrtlmix = 1 if m3_marital ~= f3_marital & m3_marital ~= . & f3_marital ~= . ;
467
468 * Create couple relp duration var--average of her and his report *;
469
470 gen c3_rdur = (m3_rdur + f3_rdur) / 2 ;
471 sum *rdur ;
472 replace c3_rdur = f3_rdur if c3_rdur == . & f3_rdur ~= . ;
473 replace c3_rdur = m3_rdur if c3_rdur == . & m3_rdur ~= . ;
474 sum *rdur ;
475
476 * ice sometimes generates out-of-range values ;
477
478 foreach var in
479     m3_sei f3_sei
480     m4_sei f4_sei
481     m3_inc f3_inc
482     m4_inc f4_inc
483     m1_hshld_inc f1_hshld_inc
484     f3_ahpvt m3_ahpvt { ;
485 replace `var' = 0 if `var' < 0 ;
486 } ;
487
488 * ice sometimes generates out-of-range values ;
489
490 foreach var in
491     m3_sei f3_sei

```



```

492     m4_sei f4_sei { ;
493     replace `var' = 96 if `var' > 96 & `var'~= . ;
494     } ;
495
496     * imputed both but really only needed to impute one and they ought to be consistent ;
497
498     replace f3_ln_inc=ln(f3_inc+1) ;
499     replace m3_ln_inc=ln(m3_inc+1) ;
500
501     * ice sometimes generates implausible values ;
502
503     replace f3_bmi = 15.5 if f3_bmi < 15 ;
504     replace m3_bmi = 13.5 if m3_bmi < 13 ;
505
506     replace f3_yrsedu = 8 if f3_yrsedu < 8 ;
507     replace m3 yrsedu = 8 if m3 yrsedu < 8 ;
508
509     * ice sometimes generates out-of-range values--all relps are at least 3 months long in
    order to be eligible ;
510
511     replace f3_rdur = 3 if f3_rdur < 3 ;
512     replace m3_rdur = 3 if m3_rdur < 3 ;
513
514     * ice sometimes generates implausible values ;
515
516     sum f3_inc m3_inc if _mj==0 ;
517     sum f3_inc m3_inc if _mj>0 ;
518     replace f3_inc=0 if f3_inc<0 ;
519     replace m3_inc=0 if m3_inc<0 ;
520     replace f3_inc=300000 if f3_inc>300000 & f3_inc~= . ;
521     replace m3_inc=300000 if m3_inc>300000 & m3_inc~= . ;
522     sum f3_inc m3_inc if _mj>0 ;
523
524     gen f3_underweight = 0 ; replace f3_underweight = 1 if f3_bmi < 18.5 ;
525     gen f3_normalweight = 0 ; replace f3_normalweight = 1 if f3_bmi >= 18.5 & f3_bmi < 25 ;
526     gen f3_overweight = 0 ; replace f3_overweight = 1 if f3_bmi >= 25 & f3_bmi < 30 ;
527     gen f3_obese = 0 ; replace f3_obese = 1 if f3_bmi > 30 ;
528     gen f3_weightcat = f3_underweight ;
529     replace f3_weightcat = 2 if f3_normalweight == 1 ;
530     replace f3_weightcat = 3 if f3_overweight == 1 ;
531     replace f3_weightcat = 4 if f3_obese == 1 ;
532     tab f3_weightcat ; sum f3_underweight f3_normalweight f3_overweight f3_obese ;
533
534     gen m3_underweight = 0 ; replace m3_underweight = 1 if m3_bmi < 18.5 ;
535     gen m3_normalweight = 0 ; replace m3_normalweight = 1 if m3_bmi >= 18.5 & m3_bmi < 25 ;
536     gen m3_overweight = 0 ; replace m3_overweight = 1 if m3_bmi >= 25 & m3_bmi < 30 ;
537     gen m3_obese = 0 ; replace m3_obese = 1 if m3_bmi > 30 ;
538     gen m3_weightcat = m3_underweight ;
539     replace m3_weightcat = 2 if m3_normalweight == 1 ;
540     replace m3_weightcat = 3 if m3_overweight == 1 ;
541     replace m3_weightcat = 4 if m3_obese == 1 ;
542     tab m3_weightcat ; sum m3_underweight m3_normalweight m3_overweight m3_obese ;
543
544     sum *underweight *normalweight *overweight *obese ;
545
546     recode m3_physatt 1/3=0 4/5=1, gen(m3_attvatt) ;
547     recode f3_physatt 1/3=0 4/5=1, gen(f3_attvatt) ;
548     sum m3_attvatt f3_attvatt ;
549
550     *tab f3_marital m3_marital ;
551     tab f3_marital f3_married ;
552     tab f3_marital f3_cohab ;
553     tab f3_marital f3_dating ;
554     tab m3_marital m3_married ;
555     tab m3_marital m3_cohab ;
556     tab m3_marital m3_dating ;
557
558     alpha f3_physatt f3_groomed f3_peratt, item gen(f3_ovatt) ;
559     label var f3_ovatt "index of interviewer evaluation, higher=more desirable, a=0.78" ;
560     alpha m3_physatt m3_groomed m3_peratt, item gen(m3_ovatt) ;

```

```

561 label var m3_ovatt "index of interviewer evaluation, higher=more desirable, a=0.76" ;
562
563 alpha f3_physatt f3_groomed, item gen(f3_ovatt2) ;
564 label var f3_ovatt2 "index of interviewer evaluation, higher=more desirable, a=0.??" ;
565 alpha m3_physatt m3_groomed, item gen(m3_ovatt2) ;
566 label var m3_ovatt2 "index of interviewer evaluation, higher=more desirable, a=0.??" ;
567
568 alpha f3_physatt f3_underweight f3_overweight f3_obese f3_groomed f3_peratt, item gen(
f3_ovatt_wgtgrp) ;
569 label var f3_ovatt_wgtgrp "index of interviewer evaluation & weight group, higher=more
desirable, a=0.62" ;
570 alpha m3_physatt f3_underweight f3_overweight f3_obese m3_groomed m3_peratt, item gen(
m3_ovatt_wgtgrp) ;
571 label var m3_ovatt_wgtgrp "index of interviewer evaluation & weight group, higher=more
desirable, a=0.59" ;
572
573 alpha f3_physatt f3_weightcat f3_groomed f3_peratt, item gen(f3_ovatt_wgtgrp2) ;
574 label var f3_ovatt_wgtgrp2 "index of interviewer evaluation & weight group, higher=more
desirable, a=0.??" ;
575 alpha m3_physatt f3_weightcat m3_groomed m3_peratt, item gen(m3_ovatt_wgtgrp2) ;
576 label var m3_ovatt_wgtgrp2 "index of interviewer evaluation & weight group, higher=more
desirable, a=0.??" ;
577
578 * SES index * ;
579 alpha
580 m3_yrsedu m3_sei m3_npboss90 m3_hh7,
581 item gen(m3_sesidx) ;
582 alpha
583 f3_yrsedu f3_sei f3_npboss90 f3_hh7,
584 item gen(f3_sesidx) ;
585
586 * Difference measures--his minus her--used in Table 4 and Table 6 ;
587
588 foreach var in
589 bmi
590 sesidx
591 physatt groomed
592 yrsedu ln_income ahpvt
593 peratt emosup
594 calcage3
595 ee cgrdp
596 ovatt ovatt_wgtgrp
597 sei npboss90 { ;
598 gen mf3_`var' = m3_`var' - f3_`var' ;
599 gen fm3_`var' = f3_`var' - m3_`var' ;
600 pwcorr mf3_`var' fm3_`var' ;
601 } ;
602
603 * This is wierd for ln_income because ln(x) - ln(y) = ln(x/y). So it is a log of a ratio as
much as it is a difference of logs. But taking the difference and then logging it doesn't
work either because the log is undefined for n <= 0. ;
604 gen mf3_income = exp(m3_ln_income) - exp(f3_ln_income) ;
605 gen fm3_income = exp(f3_ln_income) - exp(m3_ln_income) ;
606
607 sum fm3* mf3*, det ;
608
609 recode f3_yrsedu 0/11=1 12=2 13/15=3 16=4 17/99=5, gen(f3_edu5) ;
610 recode m3_yrsedu 0/11=1 12=2 13/15=3 16=4 17/99=5, gen(m3_edu5) ;
611 label define edu5
612 1 "lt HS"
613 2 "HS Graduate"
614 3 "some College"
615 4 "College Grad"
616 5 "Grad/Prof Degree",
617 modify ;
618 label values f3 edu5 edu5 ;
619 label values m3_edu5 edu5 ;
620
621 table f3_edu5 if _mj==0, contents(mean mf3_bmi mean mf3_physatt mean mf3_groomed) ;
622 table m3_edu5 if _mj==0, contents(mean mf3_bmi mean mf3_physatt mean mf3_groomed) ;

```

```

623 table f3_edu5 if _mj==0, contents(mean mf3_ln_income mean mf3_ahpvt mean mf3_calcage3) ;
624 table m3_edu5 if _mj==0, contents(mean mf3_ln_income mean mf3_ahpvt mean mf3_calcage3) ;
625 table f3_edu5 if _mj==0, contents(mean mf3_peratt mean mf3_emosup) ;
626 table m3_edu5 if _mj==0, contents(mean mf3_peratt mean mf3_emosup) ;
627
628 table f3_physatt if _mj==0, contents(mean mf3_bmi mean mf3_physatt mean mf3_groomed) ;
629 table m3_physatt if _mj==0, contents(mean mf3_bmi mean mf3_physatt mean mf3_groomed) ;
630 table f3_physatt if _mj==0, contents(mean mf3_ln_income mean mf3_ahpvt mean mf3_calcage3) ;
631 table m3_physatt if _mj==0, contents(mean mf3_ln_income mean mf3_ahpvt mean mf3_calcage3) ;
632 table f3_physatt if _mj==0, contents(mean mf3_peratt mean mf3_emosup) ;
633 table m3_physatt if _mj==0, contents(mean mf3_peratt mean mf3_emosup) ;
634
635 recode f3_physatt 1/2=2 4/5=4, gen(f3_physatt3) ;
636 recode m3_physatt 1/2=2 4/5=4, gen(m3_physatt3) ;
637
638 table f3_physatt3 if _mj==0, contents(mean mf3_bmi mean mf3_physatt mean mf3_groomed) ;
639 table m3_physatt3 if _mj==0, contents(mean mf3_bmi mean mf3_physatt mean mf3_groomed) ;
640 table f3_physatt3 if _mj==0, contents(mean mf3_ln_income mean mf3_ahpvt mean mf3_calcage3) ;
641 table m3_physatt3 if _mj==0, contents(mean mf3_ln_income mean mf3_ahpvt mean mf3_calcage3) ;
642 table f3_physatt3 if _mj==0, contents(mean mf3_peratt mean mf3_emosup) ;
643 table m3_physatt3 if _mj==0, contents(mean mf3_peratt mean mf3_emosup) ;
644
645 sum m3_rdur, det ;
646 egen m3_rdur1 = pctlile(m3_rdur), p(33) ;
647 egen m3_rdur2 = pctlile(m3_rdur), p(66) ;
648 gen m3_rdur_short = 0 ;
649 gen m3_rdur_med = 0 ;
650 gen m3_rdur_long = 0 ;
651 replace m3_rdur_short = 1 if m3_rdur <= m3_rdur1 ;
652 replace m3_rdur_med = 1 if m3_rdur > m3_rdur1 & m3_rdur <= m3_rdur2 ;
653 replace m3_rdur_long = 1 if m3_rdur >= m3_rdur2 ;
654 sum m3_rdur* ; drop m3_rdur1 m3_rdur2 ;
655
656 sum f3_rdur, det ;
657 egen f3_rdur1 = pctlile(f3_rdur), p(33) ;
658 egen f3_rdur2 = pctlile(f3_rdur), p(66) ;
659 gen f3_rdur_short = 0 ;
660 gen f3_rdur_med = 0 ;
661 gen f3_rdur_long = 0 ;
662 replace f3_rdur_short = 1 if f3_rdur <= f3_rdur1 ;
663 replace f3_rdur_med = 1 if f3_rdur > f3_rdur1 & f3_rdur <= f3_rdur2 ;
664 replace f3_rdur_long = 1 if f3_rdur >= f3_rdur2 ;
665 sum f3_rdur* ; drop f3_rdur1 f3_rdur2 ;
666
667 sum c3_rdur, det ;
668 egen c3_rdur1 = pctlile(c3_rdur), p(33) ;
669 egen c3_rdur2 = pctlile(c3_rdur), p(66) ;
670 gen c3_rdur_short = 0 ;
671 gen c3_rdur_med = 0 ;
672 gen c3_rdur_long = 0 ;
673 replace c3_rdur_short = 1 if c3_rdur <= c3_rdur1 ;
674 replace c3_rdur_med = 1 if c3_rdur > c3_rdur1 & c3_rdur <= c3_rdur2 ;
675 replace c3_rdur_long = 1 if c3_rdur >= c3_rdur2 ;
676 sum c3_rdur* ; drop c3_rdur1 c3_rdur2 ;
677
678 *** Occ Status Diff Vars ***;
679
680 * diff between Partners' Occ Statuses *;
681
682 pwcorr m3_hh7 f3_hh7, sig ;
683 gen hh7_diff = m3_hh7 - f3_hh7 ;
684 label var hh7_diff "m3_hh7 - f3_hh7" ;
685 sum hh7_diff ;
686
687 pwcorr m4_hh7 f4_hh7, sig ;
688 gen hh7_diff4 = m4_hh7 - f4_hh7 ;
689 label var hh7_diff4 "m4_hh7 - f4_hh7" ;
690 sum hh7_diff4 ;
691
692 * diff between male partner's occ status and female partner's dad's occ status *;

```

```

693
694 pwcorr m3_hh7 f1_dad_hh7, sig ;
695 gen hh7_m3d_diff = m3_hh7 - f1_dad_hh7 ;
696 label var hh7_m3d_diff "m3_hh7 - f1_dad_hh7" ;
697 sum hh7_m3d_diff ;
698
699 pwcorr m3_npboss90 f1_dad_npboss90, sig ;
700 gen npboss90_m3d_diff = m3_npboss90 - f1_dad_npboss90 ;
701 label var npboss90_m3d_diff "m3_npboss90 - f1_dad_npboss90" ;
702 sum npboss90_m3d_diff ;
703
704 * diff between female partner's occ status and male partner's dad's occ status *;
705
706 pwcorr f3_hh7 m1_dad_hh7, sig ;
707 gen hh7_f3d_diff = f3_hh7 - m1_dad_hh7 ;
708 label var hh7_f3d_diff "f3_hh7 - m1_dad_hh7" ;
709 sum hh7_f3d_diff ;
710
711 pwcorr f3_npboss90 m1_dad_npboss90, sig ;
712 gen npboss90_f3d_diff = f3_npboss90 - m1_dad_npboss90 ;
713 label var npboss90_f3d_diff "f3_npboss90 - m1_dad_npboss90" ;
714 sum npboss90_f3d_diff ;
715
716 * diff between male partner's occ status and female partner's mom's occ status *;
717
718 pwcorr m3_hh7 f1_mom_hh7, sig ;
719 gen hh7_m3m_diff = m3_hh7 - f1_mom_hh7 ;
720 label var hh7_m3m_diff "m3_hh7 - f1_mom_hh7" ;
721 sum hh7_m3m_diff ;
722
723 * diff between female partner's occ status and male partner's mom's occ status *;
724
725 pwcorr f3_hh7 m1_mom_hh7, sig ;
726 gen hh7_f3m_diff = f3_hh7 - m1_mom_hh7 ;
727 label var hh7_f3m_diff "f3_hh7 - m1_mom_hh7" ;
728 sum hh7_f3m_diff ;
729
730 * diff between male's dad's occ status and female's dad's occ status *;
731
732 pwcorr f1_dad_hh7 m1_dad_hh7, sig ;
733 gen hh7_dad_diff = f1_dad_hh7 - m1_dad_hh7 ;
734 label var hh7_dad_diff "f1_dad_hh7 - m1_dad_hh7" ;
735 sum hh7_dad_diff ;
736
737 pwcorr f1_dad_npboss90 m1_dad_npboss90, sig ;
738 gen npboss90_dad_diff = f1_dad_npboss90 - m1_dad_npboss90 ;
739 label var npboss90_dad_diff "f1_dad_npboss90 - m1_dad_npboss90" ;
740 sum npboss90_dad_diff ;
741
742 * diff between male's mom's occ status and female's mom's occ status *;
743
744 pwcorr f1_mom_hh7 m1_mom_hh7, sig ;
745 gen hh7_mom_diff = f1_mom_hh7 - m1_mom_hh7 ;
746 label var hh7_mom_diff "f1_mom_hh7 - m1_mom_hh7" ;
747 sum hh7_mom_diff ;
748
749 * diff between male partner's w4 occ status and female partner's dad's occ status *;
750
751 pwcorr m4_hh7 f1_dad_hh7, sig ;
752 gen hh7_m4d_diff = m4_hh7 - f1_dad_hh7 ;
753 label var hh7_m4d_diff "m4_hh7 - f1_dad_hh7" ;
754 sum hh7_m4d_diff ;
755
756 pwcorr m4_npboss90 f1_dad_npboss90, sig ;
757 gen npboss90_m4d_diff = m4_npboss90 - f1_dad_npboss90 ;
758 label var npboss90_m4d_diff "m4_npboss90 - f1_dad_npboss90" ;
759 sum npboss90_m4d_diff ;
760
761 * diff between female partner's w4 occ status and male partner's dad's occ status *;
762

```

```

763 pwcorr f4_npboss90 m1_dad_npboss90, sig ;
764 gen npboss90_f4d_diff = f4_npboss90 - m1_dad_npboss90 ;
765 label var npboss90_f4d_diff "f4_npboss90 - m1_dad_npboss90" ;
766 sum npboss90_f4d_diff ;
767
768 pwcorr f4_hh7 m1_dad_hh7, sig ;
769 gen hh7_f4d_diff = f4_hh7 - m1_dad_hh7 ;
770 label var hh7_f4d_diff "f4_hh7 - m1_dad_hh7" ;
771 sum hh7_f4d_diff ;
772
773 * diff between male partner's w4 occ status and female partner's mom's occ status *;
774
775 pwcorr m4_hh7 f1_mom_hh7, sig ;
776 gen hh7_m4m_diff = m4_hh7 - f1_mom_hh7 ;
777 label var hh7_m4m_diff "m4_hh7 - f1_mom_hh7" ;
778 sum hh7_m4m_diff ;
779
780 * diff between female partner's w4 occ status and male partner's mom's occ status *;
781
782 pwcorr f4_hh7 m1_mom_hh7, sig ;
783 gen hh7_f4m_diff = f4_hh7 - m1_mom_hh7 ;
784 label var hh7_f4m_diff "f4_hh7 - m1_mom_hh7" ;
785 sum hh7_f4m_diff ;
786
787 * diff between female parnter's mobility and male partner's mobility -- hh7*;
788
789 pwcorr hh7_m3d_diff hh7_f3d_diff ;
790 gen mf3_hh7_mob_diff = hh7_m3d_diff - hh7_f3d_diff ;
791 label var mf3_hh7_mob_diff "hh7_m3d_diff - hh7_f3d_diff" ;
792 sum mf3_hh7_mob_diff ;
793
794 * diff between female parnter's mobility and male partner's mobility -- npboss90 *;
795
796 pwcorr npboss90_m3d_diff npboss90_f3d_diff ;
797 gen mf3_npb_mob_diff = npboss90_m3d_diff - npboss90_f3d_diff ;
798 label var mf3_npb_mob_diff "npboss90_m3d_diff - npboss90_f3d_diff" ;
799 sum mf3_npb_mob_diff ;
800
801 des hh7_diff
802     hh7_m3d_diff hh7_f3d_diff hh7_m3m_diff hh7_f3m_diff
803     hh7_dad_diff hh7_mom_diff
804     hh7_m4d_diff hh7_f4d_diff hh7_m4m_diff hh7_f4m_diff ;
805 sum hh7_diff
806     hh7_m3d_diff hh7_f3d_diff hh7_m3m_diff hh7_f3m_diff
807     hh7_dad_diff hh7_mom_diff
808     hh7_m4d_diff hh7_f4d_diff hh7_m4m_diff hh7_f4m_diff ;
809
810 /*
811 -- when female was the recruited partner (f3_partner==1) then there is w4 info on the male
partner.
812 -- when female was not the recruited partner (f3_partner==0) then there is w4 info on the
female partner.
813 -- when female was not the recruited partner (f3_partner==0) then there is not w4 info on
the male partner.
814 -- when female was the recruited partner (f3_partner==1) then there is not w4 info on the
female partner.
815 */
816
817 foreach var in m4_hh7 hh7_m4d_diff hh7_m4m_diff m4_npboss90 npboss90_m4d_diff { ;
818
819 replace `var' = . if f3_partner == 0 ;
820
821 } ;
822
823 foreach var in f4_hh7 hh7_f4d_diff hh7_f4m_diff f4_npboss90 npboss90_f4d_diff { ;
824
825 replace `var' = . if f3_partner == 1 ;
826
827 } ;
828

```

```
829 * Average couple age--I examine age diff too but most couples are similar in age ;
830
831 gen c3_calcage3 = (m3_calcage3 + f3_calcage3)/2 ;
832
833 ** Family of origins (Dad) working class or white collar? **;
834
835 * Prior authors (eg Elder) thought that women of working-class origins would be more likely
to use beauty to achieve upward mobility ;
836
837 foreach ltr in f m { ;
838
839 gen `ltr'1_wc_hh7 = 0 ;
840 replace `ltr'1_wc_hh7 = 1 if `ltr'1_dad_hh7 >=4 ;
841 label var `ltr'1_wc_hh7 "working class origins, HH7" ;
842
843 gen `ltr'1_wc_sei = 0 ;
844 replace `ltr'1_wc_sei = 1 if `ltr'1_dad_sei <= 50 ;
845 label var `ltr'1_wc_sei "working class origins, sei" ;
846
847 gen `ltr'1_wc_npboss90 = 0 ;
848 replace `ltr'1_wc_npboss90 = 1 if `ltr'1_dad_npboss90 <= 50 ;
849 label var `ltr'1_wc_npboss90 "working class origins, npboss90" ;
850
851 } ;
852
853 *** Save Data for Analysis in "exchange and matching - analysis" ***;
854
855 save "...\\program1--prepare data.dta", replace ;
856
857 *** End Program ***;
858
859 clear ; log close ;
860
```