

```

1 *****
2 ***SYNTAX FOR "Socio-economic inequalities in physical activity among older adults before and during
the COVID-19 pandemic: Evidence from the English Longitudinal Study of Ageing"***
3 *****
4
5 * STATA version: 17.0, BE-Basic Edition
6
7 * STATA citation: StataCorp. 2021. Stata Statistical Software: Release 17. College Station, TX:
StataCorp LLC.
8
9 * Data citation (main ELSA survey): Banks, J., Batty, G. David, Breedvelt, J., Coughlin, K.,
Crawford, R., Marmot, M., Nazroo, J., Oldfield, Z., Steel, N., Steptoe, A., Wood, M., Zaninotto, P.
(2021). English Longitudinal Study of Ageing: Waves 0-9, 1998-2019. [data collection]. 37th Edition.
UK Data Service. SN: 5050, DOI: 10.5255/UKDA-SN-5050-24
10
11 * Data citation (COVID-19 sub-study): Steptoe, A., Addario, G., Banks, J., Batty, G. David,
Coughlin, K., Crawford, R., Dangerfield, P., Marmot, M., Nazroo, J., Oldfield, Z., Pacchiotti, B.,
Steel, N., Wood, M., Zaninotto, P. (2021). English Longitudinal Study of Ageing COVID-19 Study,
Waves 1-2, 2020. [data collection]. 2nd Edition. UK Data Service. SN: 8688, DOI:
10.5255/UKDA-SN-8688-2
12
13 * Data access statement: ELSA data from the main survey (SN 5050) and the COVID-19 sub-study (SN
8688) are available through the UK Data Service (https://ukdataservice.ac.uk/). The main ELSA
dataset is safeguarded and can be accessed via
https://beta.ukdataservice.ac.uk/datacatalogue/studies/study?id=5050#!/access-data. The COVID-19
sub-study can be accessed via
https://beta.ukdataservice.ac.uk/datacatalogue/studies/study?id=8688#!/access-data. More information
on how to access ELSA, including the conditions of use, can be found on the UK Data Service website
(main ELSA survey: https://beta.ukdataservice.ac.uk/datacatalogue/studies/study?id=5050#!/details;
COVID-19 sub-study: https://beta.ukdataservice.ac.uk/datacatalogue/studies/study?id=8688#!/details)
and the ELSA website (main ELSA survey: https://www.elsa-project.ac.uk/accessing-elsa-data; COVID-19
sub-study: https://www.elsa-project.ac.uk/covid-19-data).
14
15 * Date of data access/download (dd/mm/yyyy): 17/12/2021
16
17 * Project ID: 217429
18
19 * Data documentation: Documentation pertaining to ELSA (e.g., data dictionaries, questionnaires,
technical reports, user guides) is available on the UK Data Service website (main ELSA survey:
https://beta.ukdataservice.ac.uk/datacatalogue/studies/study?id=5050#!/documentation; COVID-19
sub-study: https://beta.ukdataservice.ac.uk/datacatalogue/studies/study?id=8688#!/documentation) and
the ELSA website (main ELSA survey: https://www.elsa-project.ac.uk/data-and-documentation; COVID-19
sub-study: https://www.elsa-project.ac.uk/covid-19-data).
20
21 *****
22 ***DATA PROCESSING***
23 *****
24
25 * Change working directory - add pathname in between quotation marks for Windows
26 cd ""
27
28 * Variables Wave 9
29 use idauniq heacta heactb heactc w9nssec8 w9nssec3 samptyp w9xwgt w9scwt indsex indager dimarr
fqethnmr wpdes hhtot heill helim hehelp psceda pscedb pscedc pscedd pscede pscedf pscedg pscedh
scalcm hesmk heska heskd heske heskf hestop heskb heskc hecgstp hecgsta using wave_9_elsa_data_eul_v1
.dta
30 * Describe dataset
31 describe
32 * Sort from lowest to highest participant identifier (ID)
33 sort idauniq
34 * Rename variables to shorter forms

```

```

35  rename w9nssec8 nssec8
36  rename w9nssec3 nssec3
37  rename indsex Sex
38  * Generate a new variable called wave and assign the number 9 to each observation (to designate Wave
    9)
39  gen wave = 9
40  * Save Wave 9 core dataset
41  save wave9par.dta
42
43  * Variables COVID Wave 2
44  use idauniq HEACTA HEACTB HEACTC Finstat_w1 Cohort CorePartner CvNumP CvMhCed_CvMhCed1_q
    CvMhCed_CvMhCed2_q CvMhCed_CvMhCed3_q CvMhCed_CvMhCed4_q CvMhCed_CvMhCed5_q CvMhCed_CvMhCed6_q
    CvMhCed_CvMhCed7_q CvMhCed_CvMhCed8_q CvVulnB CvStayD1 CvStayD2 CvStayD3 CvStayD4 CvStayD5 CvHesmoke
    HESKB HESKC CvPred CvPstd CvHeSelf RelStat HEILL HELIM Sex Age_arch Ethnicity_arch wtfin1 wtfin2
    cov19lwtw2 cov19lwtw2b cov19lwtw2c using elsa_covid_w2_eul.dta
45  * Describe dataset
46  describe
47  * Sort from lowest to highest participant ID
48  sort idauniq
49  * Rename variables to shorter forms and to ensure consistency with Wave 9 (for heacta-heactc)
50  rename HEACTA heacta
51  rename HEACTB heactb
52  rename HEACTC heactc
53  rename Finstat_w1 FinStat
54  rename CvNumP hhtotw2
55  rename CvMhCed_CvMhCed1_q pscedaw2
56  rename CvMhCed_CvMhCed2_q pscedbw2
57  rename CvMhCed_CvMhCed3_q pscedcw2
58  rename CvMhCed_CvMhCed4_q psceddw2
59  rename CvMhCed_CvMhCed5_q pscedew2
60  rename CvMhCed_CvMhCed6_q pscedfw2
61  rename CvMhCed_CvMhCed7_q pscedgw2
62  rename CvMhCed_CvMhCed8_q pscedhw2
63  rename HEILL heillw2
64  rename HELIM helimw2
65  rename Sex Sexw2
66  * Generate a new variable called wave and assign the number 11 to each observation (to designate
    COVID Wave 2)
67  gen wave = 11
68  * Save COVID Wave 2 core dataset
69  save covidwave2par.dta
70
71  * Variables Wave 9 Derived
72  use idauniq edqual using wave_9_ifs_derived_variables.dta
73  * Describe dataset
74  describe
75  * Sort from lowest to highest participant ID
76  sort idauniq
77  * Save Wave 9 derived dataset
78  save wave9derivedr.dta
79
80  * Variables Wave 9 Financial Derived
81  use idauniq nettotw_bu_s nettotw_bu_f nettotw_bu_t totwq5_bu_s using
    wave_9_financial_derived_variables.dta
82  * Describe dataset
83  describe
84  * Sort from lowest to highest participant ID
85  sort idauniq
86  * Save Wave 9 financial dataset
87  save wave9financialr.dta
88
89  * Wave 9 complete data
90  * Merge core, derived, and financial datasets for Wave 9 using the participant ID

```

```

91 * Use Wave 9 core dataset
92 use wave9par.dta
93 * One-to-one merge of data in memory with wave9financialr.dta on participant ID
94 merge 1:1 idauniq using wave9financialr.dta, generate (merge_financial9)
95 * Overwrite Wave 9 dataset, by replacing the previously saved file
96 save wave9par.dta, replace
97 * Use the newly saved file for Wave 9
98 use wave9par.dta
99 * One-to-one merge of data in memory with wave9derivedr.dta on participant ID
100 merge 1:1 idauniq using wave9derivedr.dta, generate (merge_derived9)
101 * Sort from lowest to highest participant ID
102 sort idauniq
103 * Overwrite Wave 9 dataset, by replacing the previously saved file
104 save wave9par.dta, replace
105
106 * Append Wave 9 and COVID Wave 2 datasets
107 use wave9par.dta
108 append using covidwave2par.dta
109 * Sort by participant ID and wave (lowest to highest)
110 sort idauniq wave
111 * Assigns a number in ascending order to each row of observations
112 gen ascnr = _n
113
114 * Unique individual serial number (personal ID)
115 * Replace variable as missing for any missing cases (coded as negative numbers in the ELSA dataset)
116 replace idauniq = . if idauniq<0
117
118 * Organising dataset
119 * Generate a variable that assigns the observation number (i.e., 1 for first data collection
120 timepoint, 2 for second data collection timepoint) to each row by participant ID
121 bysort idauniq (wave): gen obsnr = _n
122 * Generate a variable that assigns the number of total observations to each row of data for a given
123 participant
124 bysort idauniq: gen obscount = _N
125 * Check how many participants have data at 1 or 2 timepoints - the "if obsnr==1" statement is used
126 to prevent participants with data at two timepoints from contributing to the counts twice
127 tabulate obscount if obsnr==1
128 * Generate a variable that assigns the number 1 to the row representing participants' first
129 observation
130 bysort idauniq (wave): gen first = 1 if _n==1
131 * Generate a variable that assigns the number 1 to the row representing participants' last observation
132 bysort idauniq (wave): gen last = 1 if _n==_N
133 * Generate a variable that assigns the number 1 to the row representing participants' first
134 observation if this corresponds to Wave 9 (baseline)
135 bysort idauniq (wave): gen firstwave = 1 if obsnr==1 & wave==9
136 * Carry the value of this last variable forwards to the remainder of a participant's observations
137 bysort idauniq: gen variable = firstwave[1]
138 * Install unique command
139 ssc install unique
140 * Count total number of participants and observations
141 unique idauniq
142 * 9,014 individuals, 15,530 observations
143 unique idauniq if wave==9
144 * 8,736 individuals
145 unique idauniq if wave==11
146 * 6,794 individuals
147 * Check types of sample members
148 tab samptyp
149 tab FinStat
150 * Assign the COVID Wave 2 longitudinal weight to all observations for a participant
151 bysort idauniq(wave): replace cov19lwgtw2 = cov19lwgtw2[2]
152 * Drop if participant is not a core member (i.e., if they do not have a valid sampling weight
153 assigned)

```

```

148 drop if inlist(cov19lwgtw2,-1,.)
149 * Count total number of participants and observations
150 unique idauniq
151 * 5,378 individuals, 10,756 observations
152 * Replace age = 90 if participant is aged 90+ years (collapsed in ELSA and coded as -7 at Wave 9)
153 replace indager = 90 if indager== -7
154
155 * Generate a new variable duplicating the age variable at Wave 9
156 gen indager_cons = indager if wave==9
157 * Declare a panel dataset with participant ID "idauniq" and time variable "wave"
158 tsset idauniq wave
159 * Install carryforward command
160 ssc install carryforward
161 * Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID
Wave 2) by participant ID
162 bysort idauniq: carryforward indager_cons, replace
163 * Drop observation if the participant is aged less than 60 years at Wave 9
164 drop if indager_cons < 60
165 * Count total number of participants and observations
166 unique idauniq
167 * 4,407 individuals, 8,814 observations
168 * Save dataset with a new name
169 save datapar.dta
170
171 * Vigorous/Moderate/Mild sports or activities (Wave 9, COVID Wave 2)
172 * Replace variables as missing for any missing cases (coded as negative numbers in the ELSA dataset)
173 replace heacta = . if heacta<0
174 replace heactb = . if heactb<0
175 replace heactc = . if heactc<0
176 * Generate a new variable
177 gen activity2 = .
178 * Assign the number 3 if the participant partakes in vigorous activity more than once a week or
("|") once a week
179 replace activity2 = 3 if heacta==1 | heacta==2
180 * Assign the number 2 if the participant partakes in moderate activity more than once a week or once
a week, and takes part in vigorous activity less than once a week
181 replace activity2 = 2 if (heactb==1 | heactb==2) & inlist(heacta,3,4)
182 * Assign the number 1 if the participant partakes in mild activity more than once a week or once a
week, and takes part in moderate and vigorous activities less than once a week
183 replace activity2 = 1 if (heactc==1 | heactc==2) & inlist(heacta,3,4) & inlist(heactb,3,4)
184 * Assign the number 0 if the participant does not take part in activity of any intensity once a week
or more
185 replace activity2 = 0 if inlist(heacta,3,4) & inlist(heactb,3,4) & inlist(heactc,3,4)
186 * Replace the variable as missing for participants with missing cases on all three variables
187 replace activity2 = . if inlist(heacta,.) & inlist(heactb,.) & inlist(heactc,.)
188 * Coding of final physical activity variable:
189 * 3: Vigorous activity at least once per week
190 * 2: At least moderate but no vigorous activity at least once per week
191 * 1: Only mild activity at least once per week
192 * 0: Inactive (no activity on a weekly basis)
193
194 * Highest Educational Qualification (Wave 9)
195 * Excluded foreign/other
196 * Replace variable as missing for any missing cases (coded as negative numbers in the ELSA dataset)
197 replace edqual = . if edqual<0
198 * Check participant counts in each category at Wave 9
199 tab edqual if wave==9
200 * Generate a new variable
201 gen education = .
202 * Assign the number 0 if the participant does not have any formal qualifications
203 replace education = 0 if edqual == 7
204 * Assign the number 1 if the participant has A level equivalent, 0 level equivalent, or other grade
equivalent

```

```

205 replace education = 1 if inlist(edqual,3,4,5)
206 * Assign the number 2 if the participant has completed some higher education (below degree), or has
a degree or equivalent
207 replace education = 2 if inlist(edqual,1,2)
208 * Coding of final education variable:
209 * 0: No formal qualifications
210 * 1: School qualifications
211 * 2: Higher education
212
213 * NS-SEC 8 and 3 category classification (Wave 9)
214 * Excluded Never worked and long-term unemployed
215 * Replace variables as missing for any missing cases (coded as negative numbers or 99 in the ELSA
dataset)
216 replace nssec8 = . if nssec8<0
217 replace nssec8 = . if nssec8 == 99
218 replace nssec3 = . if nssec3<0
219 replace nssec3 = . if nssec3 == 99
220 * Generate a new variable
221 gen mynssec3 = .
222 * Assign the number 2 if the participant's current or most recent occupation was coded as: Higher
managerial, administrative and professional occupations; or Lower managerial, administrative and
professional occupations
223 replace mynssec3 = 2 if inlist(nssec8,1,2)
224 * Assign the number 1 if the participant's current or most recent occupation was coded as:
Intermediate occupation; or Small employers and own account workers
225 replace mynssec3 = 1 if inlist(nssec8,3,4)
226 * Assign the number 0 if the participant's current or most recent occupation was coded as: Lower
supervisory and technical occupations; or Semi-routine occupations; or Routine occupations
227 replace mynssec3 = 0 if inlist(nssec8,5,6,7)
228 * Coding of final occupational class variable:
229 * 0: Lower occupations
230 * 1: Intermediate occupations
231 * 2: Higher occupations
232
233 * Quintiles of BU total (non-pension) wealth (Wave 9)
234 * Replace variable as missing for any missing cases (coded as negative numbers in the ELSA dataset)
235 replace totwq5_bu_s = . if totwq5_bu_s<0
236 * BU total net (non-pension) wealth (Wave 9)
237 * Replace variable as missing for any missing cases (coded as negative numbers in the ELSA dataset)
238 replace nettotw_bu_s = . if inlist(nettotw_bu_s,-999,-998,-995)
239 * Redefine quintiles for our sample of interest
240 xtile quintile = nettotw_bu_s if nettotw_bu_s!=., n(5)
241 * Coding of final wealth variable:
242 * 1: 1st quintile (lowest)
243 * 2: 2nd quintile
244 * 3: 3rd quintile
245 * 4: 4th quintile
246 * 5: 5th quintile (highest)
247
248 * Wave 9 cross-sectional weight
249 * Replace variable as missing for any missing cases (coded as negative numbers in the ELSA dataset)
250 replace w9xwgt = . if w9xwgt<0
251
252 * ELSA Covid-19 cross-sectional weight (Core members) - COVID Wave 2
253 * Replace variable as missing for any missing cases (coded as negative numbers in the ELSA dataset)
254 replace wtfin1 = . if wtfin1<0
255
256 * ELSA Covid-19 study Wave 2 longitudinal weight (covid w2 vs ELSA w9)
257 * Replace variable as missing for any missing cases (coded as negative numbers in the ELSA dataset)
258 replace cov19lwgtw2 = . if cov19lwgtw2<0
259
260 * Biological sex (Wave 9)
261 * Replace variable as missing for any missing cases (coded as negative numbers in the ELSA dataset)

```

```

262 replace Sex = . if Sex<0
263 * Assign the number 0 if the participant is male
264 replace Sex = 0 if Sex == 1
265 * Assign the number 1 if the participant is female
266 replace Sex = 1 if Sex == 2
267 * Coding of the final biological sex variable:
268 * 0: Male, 1: Female
269
270 * Ethnicity (Wave 9)
271 * Replace variable as missing for any missing cases (coded as negative numbers in the ELSA dataset)
272 replace fqethnmr = . if fqethnmr<0
273 * Assign the number 0 if the participant is White
274 replace fqethnmr = 0 if fqethnmr == 1
275 * Assign the number 1 if the participant is Non-White
276 replace fqethnmr = 1 if fqethnmr == 2
277 * Coding of the final ethnicity variable:
278 * 0: White, 1: Non-White
279
280 * Number of people in household (Wave 9)
281 * Replace variable as missing for any missing cases (coded as negative numbers or 0 in the ELSA
dataset)
282 replace hhtot = . if hhtot<0
283 replace hhtot = . if hhtot==0
284 * Assign the number 0 if one person lives in household
285 replace hhtot = 0 if hhtot==1
286 * Assign the number 1 if more than one person lives in household
287 replace hhtot = 1 if hhtot>1 & hhtot != .
288 * Coding of the final living status variable:
289 * 0: Living alone, 1: Not living alone
290
291 * Age categorical (Wave 9)
292 * Generate a new variable
293 gen age_cat = .
294 * Assign the number 0 for participants aged 60-69 years at Wave 9
295 replace age_cat = 0 if indager >= 60 & indager <= 69
296 * Assign the number 1 for participants aged 70-79 years at Wave 9
297 replace age_cat = 1 if indager >= 70 & indager <= 79
298 * Assign the number 2 for participants aged 80+ years at Wave 9 and without missing age data
299 replace age_cat = 2 if indager >= 80 & indager != .
300 * Coding of the final categorical age variable:
301 * 0: 60-69 years
302 * 1: 70-79 years
303 * 2: 80+ years
304
305 * Limiting long-standing illness (Wave 9)
306 * Generate a new variable and assign the number 0 for participants with no long-standing illness or
a long-standing illness that is not limiting
307 gen limiting = 0 if heill == 2 | helim == 2
308 * Assign the number 1 for participants with a limiting long-standing illness
309 replace limiting = 1 if helim == 1
310 * Coding of the final limiting long-standing illness variable:
311 * 0: No long-standing illness or not limiting, 1: Limiting long-standing illness
312
313 * Save dataset with a new name
314 save data01par.dta
315
316 * Time-constant education - Wave 9
317 * Generate a new variable duplicating the education variable at Wave 9
318 gen education_cons = education if wave==9
319 * Declare a panel dataset with participant ID "idauniq" and time variable "wave"
320 tsset idauniq wave
321 * Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID
Wave 2) by participant ID

```



```

322 bysort idauniq: carryforward education_cons, replace
323
324 * Time-constant occupational class - Wave 9
325 * Generate a new variable duplicating the occupational class variable at Wave 9
326 gen mynssec3_cons = mynssec3 if wave==9
327 * Declare a panel dataset with participant ID "idauniq" and time variable "wave"
328 tsset idauniq wave
329 * Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID
Wave 2) by participant ID
330 bysort idauniq: carryforward mynssec3_cons, replace
331
332 * Time-constant wealth - Wave 9
333 * Generate a new variable duplicating the wealth variable at Wave 9
334 gen wealth_cons = quintile if wave==9
335 * Declare a panel dataset with participant ID "idauniq" and time variable "wave"
336 tsset idauniq wave
337 * Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID
Wave 2) by participant ID
338 bysort idauniq: carryforward wealth_cons, replace
339
340 * Time-constant biological sex - Wave 9
341 * Generate a new variable duplicating the biological sex variable at Wave 9
342 gen sex_cons = Sex if wave==9
343 * Declare a panel dataset with participant ID "idauniq" and time variable "wave"
344 tsset idauniq wave
345 * Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID
Wave 2) by participant ID
346 bysort idauniq: carryforward sex_cons, replace
347
348 * Time-constant ethnicity - Wave 9
349 * Generate a new variable duplicating the ethnicity variable at Wave 9
350 gen ethnicity_cons = fqethnmr if wave==9
351 * Declare a panel dataset with participant ID "idauniq" and time variable "wave"
352 tsset idauniq wave
353 * Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID
Wave 2) by participant ID
354 bysort idauniq: carryforward ethnicity_cons, replace
355
356 * Time-constant living status - Wave 9
357 * Generate a new variable duplicating the living status variable at Wave 9
358 gen living_cons = hhtot if wave==9
359 * Declare a panel dataset with participant ID "idauniq" and time variable "wave"
360 tsset idauniq wave
361 * Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID
Wave 2) by participant ID
362 bysort idauniq: carryforward living_cons, replace
363
364 * Time-constant age category - Wave 9
365 * Generate a new variable duplicating the categorical age variable at Wave 9
366 gen age_cons = age_cat if wave==9
367 * Declare a panel dataset with participant ID "idauniq" and time variable "wave"
368 tsset idauniq wave
369 * Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID
Wave 2) by participant ID
370 bysort idauniq: carryforward age_cons, replace
371
372 * Time-constant limiting long-standing illness - Wave 9
373 * Generate a new variable duplicating the limiting long-standing illness variable at Wave 9
374 gen limiting_cons = limiting if wave==9
375 * Declare a panel dataset with participant ID "idauniq" and time variable "wave"
376 tsset idauniq wave
377 * Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID
Wave 2) by participant ID

```

```

378 bysort idauniq: carryforward limiting_cons, replace
379
380 * Time variable
381 * Generate a new variable
382 gen TimePA = .
383 * Assign the number 0 for observations at Wave 9
384 replace TimePA = 0 if wave==9
385 * Assign the number 1 for observations at COVID Wave 2
386 replace TimePA = 1 if wave==11
387 * Coding of the final time variable:
388 * 0: Wave 9, 1: COVID Wave 2
389
390 * Save dataset with a new name
391 save data02par.dta
392
393 * Shielding (COVID Wave 2)
394 * Replace variables as missing for any missing cases (coded as negative numbers in the ELSA dataset)
395 replace CvStayD1 = . if CvStayD1<0
396 replace CvStayD2 = . if CvStayD2<0
397 replace CvStayD3 = . if CvStayD3<0
398 replace CvStayD4 = . if CvStayD4<0
399 replace CvStayD5 = . if CvStayD5<0
400 * Generate a new variable and assign the number 0 for participants who were neither self-isolating
    nor trying to stay at home in April
401 gen shielding = 0 if CvStayD3==1
402 * Assign the number 0 for participants who were trying to stay at home in April
403 replace shielding = 0 if CvStayD2==1
404 * Assign the number 1 for participants who were self-isolating in April
405 replace shielding = 1 if CvStayD1==1
406 * Assign the shielding value to all observations for a participant
407 bysort idauniq(wave): replace shielding = shielding[2]
408 * Coding of the final shielding variable:
409 * 0: Not shielding
410 * 1: Shielding at all times
411
412 * Self-rated health (Wave 9)
413 * Generate a new variable duplicating the self-rated health variable
414 gen health = hehelp
415 * Replace variable as missing for any missing cases (coded as negative numbers in the ELSA dataset)
416 replace health = . if health<0
417 * Reverse the self-rated health variable (this creates a new variable and adds the "rev" prefix to
    the original variable name)
418 revrs health
419 * Generate a new variable duplicating the reversed (revhealth) self-rated health variable at Wave 9
420 gen health_cons = revhealth if wave==9
421 * Declare a panel dataset with participant ID "idauniq" and time variable "wave"
422 tsset idauniq wave
423 * Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID
    Wave 2) by participant ID
424 bysort idauniq: carryforward health_cons, replace
425 * Coding of the final self-rated health variable:
426 * 1: Poor
427 * 2: Fair
428 * 3: Good
429 * 4: Very good
430 * 5: Excellent
431
432 * Depressive symptoms (Wave 9)
433 * Recode to the number 0 if participant answered "No" (items psceda-pscedc are reverse-coded)
434 replace psceda = 0 if psceda==2
435 replace pscedb = 0 if pscedb==2
436 replace pscedc = 0 if pscedc==2
437

```



```

438 * Recode to the number 0 if participant answered "Yes"
439 replace pscedd = 0 if pscedd==1
440 * Recode to the number 1 if participant answered "No"
441 replace pscedd = 1 if pscedd==2
442
443 * Recode to the number 0 if participant answered "No" (item pscede is reverse-coded)
444 replace pscede = 0 if pscede==2
445
446 * Recode to the number 0 if participant answered "Yes"
447 replace pscedf = 0 if pscedf==1
448 * Recode to the number 1 if participant answered "No"
449 replace pscedf = 1 if pscedf==2
450
451 * Recode to the number 0 if participant answered "No" (items pscedg-pscedh are reverse-coded)
452 replace pscedg = 0 if pscedg==2
453 replace pscedh = 0 if pscedh==2
454
455 * Generate new variables duplicating psceda-pscedh, but excluding missing cases (coded as negative
numbers in the ELSA dataset)
456 gen ceda = psceda if psceda>=0
457 gen cedb = pscedb if pscedb>=0
458 gen cedc = pscedc if pscedc>=0
459 gen cedd = pscedd if pscedd>=0
460 gen cede = pscede if pscede>=0
461 gen cedf = pscedf if pscedf>=0
462 gen cedg = pscedg if pscedg>=0
463 gen cedh = pscedh if pscedh>=0
464
465 * Generate a new variable equal to the sum of depressive symptoms (eight items) to create a total
depression score
466 gen depression = ceda + cedb + cedc + cedd + cede + cedf + cedg + cedh
467 * Generate a new variable duplicating the depressive symptoms variable at Wave 9
468 gen depression_cons = depression if wave==9
469 * Declare a panel dataset with participant ID "idauniq" and time variable "wave"
470 tsset idauniq wave
471 * Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID
Wave 2) by participant ID
472 bysort idauniq: carryforward depression_cons, replace
473
474 * Alcohol consumption (Wave 9)
475 * Generate a new variable and assign the number 0 if the participant's alcohol consumption was coded
as: Once or twice a month; or Once every couple of months; or Once or twice a year; or Not at all in
the last 12 months
476 gen alcohol = 0 if inlist(scalcm,5,6,7,8)
477 * Assign the number 1 if the participant's alcohol consumption was coded as: Three or four days a
week; or Once or twice a week
478 replace alcohol = 1 if inlist(scalcm,3,4)
479 * Assign the number 2 if the participant's alcohol consumption was coded as: Almost every day; or
Five or six days a week
480 replace alcohol = 2 if inlist(scalcm,1,2)
481 * Generate a new variable duplicating the alcohol consumption variable at Wave 9
482 gen alcohol_cons = alcohol if wave==9
483 * Declare a panel dataset with participant ID "idauniq" and time variable "wave"
484 tsset idauniq wave
485 * Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID
Wave 2) by participant ID
486 bysort idauniq: carryforward alcohol_cons, replace
487 * Coding of the final alcohol consumption variable:
488 * 0: Less than once a week
489 * 1: One to four times per week
490 * 2: Five or more times per week
491
492 * Smoking status (Wave 9)

```

```

493 * Generate a new variable and assign the number 0 for participants who do not smoke cigarettes at
494 all nowadays
495 gen smoking = 0 if heska==2
496 * Assign the number 0 for participants who never smoked cigarettes
497 replace smoking = 0 if hesmk==2
498 * Assign the number 1 for participants who smoke cigarettes nowadays (heska)
499 replace smoking = 1 if heska==1
500 * Assign the number 1 for participants who do smoke cigarettes nowadays (heskf)
501 replace smoking = 1 if heskf==1
502
503 * Generate a new variable duplicating the smoking status variable at Wave 9
504 gen smoking_cons = smoking if wave==9
505 * Declare a panel dataset with participant ID "idauniq" and time variable "wave"
506 tsset idauniq wave
507 * Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID
508 Wave 2) by participant ID
509 bysort idauniq: carryforward smoking_cons, replace
510 * Coding of the final smoking status variable:
511 * 0: Not currently smoking, 1: Current smoker
512
513 * Save dataset with a new name
514 save data03par.dta
515
516 * Keep variables required for analyses
517 keep idauniq wave TimePA indager edqual education_cons mynssec3_cons wealth_cons sex_cons
518 ethnicity_cons living_cons age_cons limiting_cons activity2 cov19lwgtw2 health_cons smoking_cons
519 alcohol_cons depression_cons shielding
520 * Count total number of participants and observations
521 unique idauniq
522 * 4,407 individuals, 8,814 observations
523 * Produce a table with the number of missing values and percent missing for each variable in the list
524 mdesc cov19lwgtw2 activity education_cons mynssec3_cons wealth_cons sex_cons ethnicity_cons
525 living_cons age_cons limiting_cons TimePA health_cons smoking_cons alcohol_cons depression_cons
526 shielding
527
528 * Display the correlation matrix for the variables in the dataset
529 pcorr, star(.05)
530 * Save dataset with a new name
531 save CCpadatar.dta
532
533 * Drop observation if physical activity data are missing
534 drop if activity2 == .
535 * Generate a variable that assigns the number of total observations to each row of data for a given
536 participant
537 bysort idauniq: gen obscount = _N
538 * Keep participants that have at least some data at both timepoints of interest (i.e., Wave 9 and
539 COVID Wave 2)
540 keep if obscount==2
541 * Drop unnecessary variable
542 drop obscount
543 * Count total number of participants and observations
544 unique idauniq
545 * 4,404 individuals, 8,808 observations
546 * Drop observation if education, occupational class, or wealth data are missing
547 drop if education_cons == . | mynssec3_cons == . | wealth_cons == .
548 * Generate a variable that assigns the number of total observations to each row of data for a given
549 participant
550 bysort idauniq: gen obscount = _N
551 * Keep participants that have at least some data at both timepoints of interest (i.e., Wave 9 and
552 COVID Wave 2)
553 keep if obscount==2
554 * Drop unnecessary variable
555 drop obscount

```

```

546 * Count total number of participants and observations
547 unique idauniq
548 * 3,802 individuals, 7,604 observations
549 * Drop if biological sex, ethnicity, living status, or categorical age data are missing
550 drop if sex_cons == . | ethnicity_cons == . | living_cons == . | age_cons == .
551 * Generate a variable that assigns the number of total observations to each row of data for a given
    participant
552 bysort idauniq: gen obscount = _N
553 * Keep participants that have at least some data at both timepoints of interest (i.e., Wave 9 and
    COVID Wave 2)
554 keep if obscount==2
555 * Drop unnecessary variable
556 drop obscount
557 * Count total number of participants and observations
558 unique idauniq
559 * 3,802 individuals, 7,604 observations
560 * Drop if limiting long-standing illness, depressive symptoms, or shielding data are missing
561 drop if limiting_cons == . | depression_cons == . | shielding == .
562 * Generate a variable that assigns the number of total observations to each row of data for a given
    participant
563 bysort idauniq: gen obscount = _N
564 * Keep participants that have at least some data at both timepoints of interest (i.e., Wave 9 and
    COVID Wave 2)
565 keep if obscount==2
566 * Drop unnecessary variable
567 drop obscount
568 * Count total number of participants and observations
569 unique idauniq
570 * 3,720 individuals, 7,440 observations
571
572 * Save complete case dataset with a new name
573 save CCpdatafullr.dta
574
575 * Create dummy variables for the physical activity outcome for hierarchical logistic regression
    analyses
576 * Inactive (coded as 0) versus mild, moderate, or vigorous (coded as 1)
577 gen acti0 = 1 if inlist(activity2,1,2,3)
578 replace acti0 = 0 if activity2 == 0
579 * Inactive, mild, or moderate (coded as 0) versus vigorous (coded as 1)
580 gen acti3 = 0 if inlist(activity2,0,1,2)
581 replace acti3 = 1 if activity2 == 3
582 * Save dataset with a new name
583 save ccpadummyr.dta
584
585 *****
586 ***STATISTICAL ANALYSES***
587 *****
588
589 * HIERARCHICAL LOGISTIC REGRESSION
590 * Display base levels of factor variables and their interactions in output tables
591 set showbaselevels on
592
593 * UNADJUSTED MODELS
594 * melogit: Multilevel mixed-effects logistic regression command
595 * pweight: Incorporates sampling weights at higher levels (i.e., participant level)
596 * or: Reports fixed-effects coefficients as odds ratios
597 * ##: Specifies the main effects for each variable and an interaction
598 * i.: Denotes a factor variable
599 * Model 1: Two-level logistic regression of physical activity on indicators for levels of education
    and time, and their interaction, with random intercepts by participant ID
600 melogit acti0 i.education_cons##i.TimePA || idauniq: pweight(cov19lwgtw2) or
601 melogit acti3 i.education_cons##i.TimePA || idauniq: pweight(cov19lwgtw2) or
602

```

```

603 * Model 2: Two-level logistic regression of physical activity on indicators for levels of
604 occupational class and time, and their interaction, with random intercepts by participant ID
605 melogit acti0 i.mynssec3_cons##i.TimePA || idauniq:, pweight(cov19lwgtw2) or
606 melogit acti3 i.mynssec3_cons##i.TimePA || idauniq:, pweight(cov19lwgtw2) or
607
608 * Model 3: Two-level logistic regression of physical activity on indicators for levels of wealth and
609 time, and their interaction, with random intercepts by participant ID
610 melogit acti0 i.wealth_cons##i.TimePA || idauniq:, pweight(cov19lwgtw2) or
611 melogit acti3 i.wealth_cons##i.TimePA || idauniq:, pweight(cov19lwgtw2) or
612
613 * Model 4: Two-level logistic regression of physical activity on indicators for levels of education,
614 occupational class, wealth, and time, including interactions between the three socio-economic
615 variables and time, with random intercepts by participant ID
616 melogit acti0 i.education_cons##i.TimePA i.mynssec3_cons##i.TimePA i.wealth_cons##i.TimePA || idauniq
617 :, pweight(cov19lwgtw2) or
618 melogit acti3 i.education_cons##i.TimePA i.mynssec3_cons##i.TimePA i.wealth_cons##i.TimePA || idauniq
619 :, pweight(cov19lwgtw2) or
620
621 * FULLY ADJUSTED MODELS (I.E., WITH COVARIATES)
622 * Model 1: Two-level logistic regression of physical activity on indicators for levels of education
623 and time, and their interaction (adjusted for covariates), with random intercepts by participant ID
624 melogit acti0 i.education_cons##i.TimePA i.sex_cons i.ethnicity_cons i.living_cons i.age_cons i.
625 limiting_cons depression_cons i.shielding || idauniq:, pweight(cov19lwgtw2) or
626 melogit acti3 i.education_cons##i.TimePA i.sex_cons i.ethnicity_cons i.living_cons i.age_cons i.
627 limiting_cons depression_cons i.shielding || idauniq:, pweight(cov19lwgtw2) or
628
629 * Model 2: Two-level logistic regression of physical activity on indicators for levels of
630 occupational class and time, and their interaction (adjusted for covariates), with random intercepts
631 by participant ID
632 melogit acti0 i.mynssec3_cons##i.TimePA i.sex_cons i.ethnicity_cons i.living_cons i.age_cons i.
633 limiting_cons depression_cons i.shielding || idauniq:, pweight(cov19lwgtw2) or
634 melogit acti3 i.mynssec3_cons##i.TimePA i.sex_cons i.ethnicity_cons i.living_cons i.age_cons i.
635 limiting_cons depression_cons i.shielding || idauniq:, pweight(cov19lwgtw2) or
636
637 * Model 3: Two-level logistic regression of physical activity on indicators for levels of wealth and
638 time, and their interaction (adjusted for covariates), with random intercepts by participant ID
639 melogit acti0 i.wealth_cons##i.TimePA i.sex_cons i.ethnicity_cons i.living_cons i.age_cons i.
640 limiting_cons depression_cons i.shielding || idauniq:, pweight(cov19lwgtw2) or
641 melogit acti3 i.wealth_cons##i.TimePA i.sex_cons i.ethnicity_cons i.living_cons i.age_cons i.
642 limiting_cons depression_cons i.shielding || idauniq:, pweight(cov19lwgtw2) or
643
644 * Model 4: Two-level logistic regression of physical activity on indicators for levels of education,
645 occupational class, wealth, and time, including interactions between the three socio-economic
646 variables and time (adjusted for covariates), with random intercepts by participant ID
647 melogit acti0 i.education_cons##i.TimePA i.mynssec3_cons##i.TimePA i.wealth_cons##i.TimePA i.sex_cons
648 i.ethnicity_cons i.living_cons i.age_cons i.limiting_cons depression_cons i.shielding || idauniq:,
649 pweight(cov19lwgtw2) or
650 melogit acti3 i.education_cons##i.TimePA i.mynssec3_cons##i.TimePA i.wealth_cons##i.TimePA i.sex_cons
651 i.ethnicity_cons i.living_cons i.age_cons i.limiting_cons depression_cons i.shielding || idauniq:,
652 pweight(cov19lwgtw2) or
653
654 clear
655
656 * GENERALISED LINEAR MIXED MODELS
657 * Use complete case dataset
658 use CCpadatafull.dta
659 * Display base levels of factor variables and their interactions in output tables
660 set showbaselevels on
661
662 * UNADJUSTED MODELS
663 * meologit: Multilevel mixed-effects ordered logistic regression command
664 * pweight: Incorporates sampling weights at higher levels (i.e., participant level)
665 * or: Reports fixed-effects coefficients as odds ratios
666 * ##: Specifies the main effects for each variable and an interaction

```

```

644 * i.: Denotes a factor variable
645 * Model 1: Two-level ordered logit regression of physical activity on indicators for levels of
education and time, and their interaction, with random intercepts by participant ID
646 meologit activity2 i.education_cons##i.TimePA || idauniq:, pweight(cov19lwgtw2) or
647 * Model 2: Two-level ordered logit regression of physical activity on indicators for levels of
occupational class and time, and their interaction, with random intercepts by participant ID
648 meologit activity2 i.mynssec3_cons##i.TimePA || idauniq:, pweight(cov19lwgtw2) or
649 * Model 3: Two-level ordered logit regression of physical activity on indicators for levels of
wealth and time, and their interaction, with random intercepts by participant ID
650 meologit activity2 i.wealth_cons##i.TimePA || idauniq:, pweight(cov19lwgtw2) or
651 * Model 4: Two-level ordered logit regression of physical activity on indicators for levels of
education, occupational class, wealth, and time, including interactions between the three
socio-economic variables and time, with random intercepts by participant ID
652 meologit activity2 i.education_cons##i.TimePA i.mynssec3_cons##i.TimePA i.wealth_cons##i.TimePA ||
idauniq:, pweight(cov19lwgtw2) or
653
654 * FULLY ADJUSTED MODELS (I.E., WITH COVARIATES)
655 * Model 1: Two-level ordered logit regression of physical activity on indicators for levels of
education and time, and their interaction (adjusted for covariates), with random intercepts by
participant ID
656 meologit activity2 i.education_cons##i.TimePA i.sex_cons i.ethnicity_cons i.living_cons i.age_cons i.
limiting_cons depression_cons i.shielding || idauniq:, pweight(cov19lwgtw2) or
657 * vce(unconditional): produces standard errors that account for the sampling variability of
covariates arising with complex survey data
658 * Predictive margins probabilities for each level of education, time, and the interaction of
education and time, for level 0 of the physical activity outcome variable (i.e., inactive), from the
fixed part of the model
659 margins i.education_cons i.TimePA i.education_cons#i.TimePA, predict (mu fixedonly outcome(0))
vsquish vce(unconditional)
660 * Predictive margins probabilities for each level of education, time, and the interaction of
education and time, for level 1 of the physical activity outcome variable (i.e., mild activity),
from the fixed part of the model
661 margins i.education_cons i.TimePA i.education_cons#i.TimePA, predict (mu fixedonly outcome(1))
vsquish vce(unconditional)
662 * Predictive margins probabilities for each level of education, time, and the interaction of
education and time, for level 2 of the physical activity outcome variable (i.e., moderate activity),
from the fixed part of the model
663 margins i.education_cons i.TimePA i.education_cons#i.TimePA, predict (mu fixedonly outcome(2))
vsquish vce(unconditional)
664 * Predictive margins probabilities for each level of education, time, and the interaction of
education and time, for level 3 of the physical activity outcome variable (i.e., vigorous activity),
from the fixed part of the model
665 margins i.education_cons i.TimePA i.education_cons#i.TimePA, predict (mu fixedonly outcome(3))
vsquish vce(unconditional)
666 * Predictive margins probabilities for each level of education, time, and the interaction of
education and time, for level 2 and 3 (collapsed) of the physical activity outcome variable (i.e.,
moderate or vigorous activity), from the fixed part of the model
667 margins i.education_cons i.TimePA i.education_cons#i.TimePA, expression(predict (mu fixedonly outcome
(2)) + predict (mu fixedonly outcome(3))) vsquish vce(unconditional)
668
669 * Model 2: Two-level ordered logit regression of physical activity on indicators for levels of
occupational class and time, and their interaction (adjusted for covariates), with random intercepts
by participant ID
670 meologit activity2 i.mynssec3_cons##i.TimePA i.sex_cons i.ethnicity_cons i.living_cons i.age_cons i.
limiting_cons depression_cons i.shielding || idauniq:, pweight(cov19lwgtw2) or
671 * vce(unconditional): produces standard errors that account for the sampling variability of
covariates arising with complex survey data
672 * Predictive margins probabilities for each level of occupational class, time, and the interaction
of occupational class and time, for level 0 of the physical activity outcome variable (i.e.,
inactive), from the fixed part of the model
673 margins i.mynssec3_cons i.TimePA i.mynssec3_cons#i.TimePA, predict (mu fixedonly outcome(0)) vsquish
vce(unconditional)
674 * Predictive margins probabilities for each level of occupational class, time, and the interaction

```



```

of occupational class and time, for level 1 of the physical activity outcome variable (i.e., mild
activity), from the fixed part of the model
675 margins i.mynssec3_cons i.TimePA i.mynssec3_cons#i.TimePA, predict (mu fixedonly outcome(1)) vsquish
vce(unconditional)
676 * Predictive margins probabilities for each level of occupational class, time, and the interaction
of occupational class and time, for level 2 of the physical activity outcome variable (i.e.,
moderate activity), from the fixed part of the model
677 margins i.mynssec3_cons i.TimePA i.mynssec3_cons#i.TimePA, predict (mu fixedonly outcome(2)) vsquish
vce(unconditional)
678 * Predictive margins probabilities for each level of occupational class, time, and the interaction
of occupational class and time, for level 3 of the physical activity outcome variable (i.e.,
vigorous activity), from the fixed part of the model
679 margins i.mynssec3_cons i.TimePA i.mynssec3_cons#i.TimePA, predict (mu fixedonly outcome(3)) vsquish
vce(unconditional)
680 * Predictive margins probabilities for each level of occupational class, time, and the interaction
of occupational class and time, for level 2 and 3 (collapsed) of the physical activity outcome
variable (i.e., moderate or vigorous activity), from the fixed part of the model
681 margins i.mynssec3_cons i.TimePA i.mynssec3_cons#i.TimePA, expression(predict (mu fixedonly outcome(2)
)) + predict (mu fixedonly outcome(3))) vsquish vce(unconditional)
682
683 * Model 3: Two-level ordered logit regression of physical activity on indicators for levels of
wealth and time, and their interaction (adjusted for covariates), with random intercepts by
participant ID
684 meologit activity2 i.wealth_cons##i.TimePA i.sex_cons i.ethnicity_cons i.living_cons i.age_cons i.
limiting_cons depression_cons i.shielding || idauniq:, pweight(cov19lwgtw2) or
685 * vce(unconditional): produces standard errors that account for the sampling variability of
covariates arising with complex survey data
686 * Predictive margins probabilities for each level of wealth, time, and the interaction of wealth and
time, for level 0 of the physical activity outcome variable (i.e., inactive), from the fixed part of
the model
687 margins i.wealth_cons i.TimePA i.wealth_cons#i.TimePA, predict (mu fixedonly outcome(0)) vsquish vce(
unconditional)
688 * Predictive margins probabilities for each level of wealth, time, and the interaction of wealth and
time, for level 1 of the physical activity outcome variable (i.e., mild activity), from the fixed
part of the model
689 margins i.wealth_cons i.TimePA i.wealth_cons#i.TimePA, predict (mu fixedonly outcome(1)) vsquish vce(
unconditional)
690 * Predictive margins probabilities for each level of wealth, time, and the interaction of wealth and
time, for level 2 of the physical activity outcome variable (i.e., moderate activity), from the
fixed part of the model
691 margins i.wealth_cons i.TimePA i.wealth_cons#i.TimePA, predict (mu fixedonly outcome(2)) vsquish vce(
unconditional)
692 * Predictive margins probabilities for each level of wealth, time, and the interaction of wealth and
time, for level 3 of the physical activity outcome variable (i.e., vigorous activity), from the
fixed part of the model
693 margins i.wealth_cons i.TimePA i.wealth_cons#i.TimePA, predict (mu fixedonly outcome(3)) vsquish vce(
unconditional)
694 * Predictive margins probabilities for each level of wealth, time, and the interaction of wealth and
time, for level 2 and 3 (collapsed) of the physical activity outcome variable (i.e., moderate or
vigorous activity), from the fixed part of the model
695 margins i.wealth_cons i.TimePA i.wealth_cons#i.TimePA, expression(predict (mu fixedonly outcome(2)) +
predict (mu fixedonly outcome(3))) vsquish vce(unconditional)
696
697 * Model 4: Two-level ordered logit regression of physical activity on indicators for levels of
education, occupational class, wealth, and time, including interactions between the three
socio-economic variables and time (adjusted for covariates), with random intercepts by participant ID
698 meologit activity2 i.education_cons##i.TimePA i.mynssec3_cons##i.TimePA i.wealth_cons##i.TimePA i.
sex_cons i.ethnicity_cons i.living_cons i.age_cons i.limiting_cons depression_cons i.shielding ||
idauniq:, pweight(cov19lwgtw2) or
699
700 * GENERALISED LINEAR LATENT AND MIXED MODELS
701
702 * Dummy variables

```



```

703 * Education
704 * Medium education (i.e., school qualifications) (coded as 1) versus low (i.e., no formal
705 qualifications) or high (i.e., higher education) education (coded as 0)
706 gen medumed = 0 if inlist(education_cons,0,2)
707 replace medumed = 1 if education_cons == 1
708 * High education (coded as 1) versus low or medium education (coded as 0)
709 gen highed = 0 if inlist(education_cons,0,1)
710 replace highed = 1 if education_cons == 2
711 * Occupational class
712 * Intermediate occupations (coded as 1) versus lower or higher occupations (coded as 0)
713 gen mediumocc = 0 if inlist(mynssec3_cons,0,2)
714 replace mediumocc = 1 if mynssec3_cons == 1
715 * Higher occupations (coded as 1) versus lower or intermediate occupations (coded as 0)
716 gen highocc = 0 if inlist(mynssec3_cons,0,1)
717 replace highocc = 1 if mynssec3_cons == 2
718 * Wealth
719 * 2nd quintile (coded as 1) versus 1st, 3rd, 4th, or 5th quintile (coded as 0)
720 gen quint2 = 0 if inlist(wealth_cons,1,3,4,5)
721 replace quint2 = 1 if wealth_cons == 2
722 * 3rd quintile (coded as 1) versus 1st, 2nd, 4th, or 5th quintile (coded as 0)
723 gen quint3 = 0 if inlist(wealth_cons,1,2,4,5)
724 replace quint3 = 1 if wealth_cons == 3
725 * 4th quintile (coded as 1) versus 1st, 2nd, 3rd, or 5th quintile (coded as 0)
726 gen quint4 = 0 if inlist(wealth_cons,1,2,3,5)
727 replace quint4 = 1 if wealth_cons == 4
728 * 5th quintile (coded as 1) versus 1st, 2nd, 3rd, or 4th quintile (coded as 0)
729 gen quint5 = 0 if inlist(wealth_cons,1,2,3,4)
730 replace quint5 = 1 if wealth_cons == 5
731
732 * Generate interaction terms
733 gen int_meded = medumed*TimePA
734 gen int_hied = highed*TimePA
735 gen int_medocc = mediumocc*TimePA
736 gen int_hiocc = highocc*TimePA
737 gen int_qui2 = quint2*TimePA
738 gen int_qui3 = quint3*TimePA
739 gen int_qui4 = quint4*TimePA
740 gen int_qui5 = quint5*TimePA
741
742 * Save dataset with a new name
743 save gllammr.dta
744
745 * UNADJUSTED MODELS
746 * gllamm: Generalised linear latent and mixed model command
747 * i(): Gives the variable that identifies the clusters
748 * link(ologit): ordinal logit link
749 * adapt: Specifies adaptive quadrature
750 * eform: Reports fixed-effects coefficients as odds ratios
751 * Model 1
752 * Estimation of proportional odds model
753 gllamm activity2 medumed highed TimePA int_meded int_hied, i(idauniq) link(ologit) pweight(
754 cov19lwgtw) adapt
755 gllamm, eform
756 estimates store model1
757 * Relax proportional odds assumption for education
758 eq thr: medumed highed
759 estimates restore model1
760 matrix a=e(b)
761 gllamm activity2 TimePA int_meded int_hied, i(idauniq) link(ologit) thresh(thr) from(a) pweight(
762 cov19lwgtw) skip adapt
763 estimates store model2
764 * Likelihood ratio test
765 lrtest model1 model2

```

```

763
764 * Model 2
765 * Estimation of proportional odds model
766 gllamm activity2 mediumocc highocc TimePA int_medocc int_hiocc, i(idauniq) link(ologit) pweight(
cov19lwgtw) adapt
767 gllamm, eform
768 estimates store model3
769 * Relax proportional odds assumption for occupational class
770 eq thr: mediumocc highocc
771 estimates restore model3
772 matrix a=e(b)
773 gllamm activity2 TimePA int_medocc int_hiocc, i(idauniq) link(ologit) thresh(thr) from(a) pweight(
cov19lwgtw) skip adapt
774 estimates store model4
775 * Likelihood ratio test
776 lrtest model3 model4
777
778 * Model 3
779 * Estimation of proportional odds model
780 gllamm activity2 quint2 quint3 quint4 quint5 TimePA int_qui2 int_qui3 int_qui4 int_qui5, i(idauniq)
link(ologit) pweight(cov19lwgtw) adapt
781 gllamm, eform
782 estimates store model5
783 * Relax proportional odds assumption for wealth
784 eq thr: quint2 quint3 quint4 quint5
785 estimates restore model5
786 matrix a=e(b)
787 gllamm activity2 TimePA int_qui2 int_qui3 int_qui4 int_qui5, i(idauniq) link(ologit) thresh(thr) from
(a) pweight(cov19lwgtw) skip adapt
788 estimates store model6
789 * Likelihood ratio test
790 lrtest model5 model6
791 clear
792
793 * Descriptive statistics
794 * Use participant ID and cross-sectional weight from Wave 9 core dataset
795 use idauniq w9xwgt using wave_9_elsa_data_eul_v1.dta
796 * Describe dataset
797 describe
798 * Sort from lowest to highest participant ID
799 sort idauniq
800 * Generate a new variable called wave and assign the number 9 to each observation (to designate Wave
9)
801 gen wave=9
802 * Save dataset with a new name
803 save wave9crossweightr.dta
804
805 * Use participant ID and cross-sectional weight from COVID Wave 2 dataset
806 use idauniq wtf1n1 using elsa_covid_w2_eul.dta
807 * Describe dataset
808 describe
809 * Sort from lowest to highest participant ID
810 sort idauniq
811 * Generate a new variable called wave and assign the number 11 to each observation (to designate
COVID Wave 2)
812 gen wave=11
813 * Save dataset with a new name
814 save wave11crossweightr.dta
815
816 * Use complete case dataset
817 use CCpdatafullr.dta
818 * One-to-one merge of data in memory with wave9crossweightr.dta on participant ID and wave
819 merge 1:1 idauniq wave using wave9crossweightr.dta, generate(merge_crossweight9)

```

```

820 * Sort from lowest to highest participant ID
821 sort idauniq
822 * Drop observations for which the key variable (participant ID) does not match
823 drop if merge_crossweight9==2
824 * Sort from lowest to highest participant ID and wave
825 sort idauniq wave
826 * One-to-one merge of data in memory with wave11crossweightr.dta on participant ID and wave
827 merge 1:1 idauniq wave using wave11crossweightr.dta, generate(merge_crossweight11)
828 * Sort from lowest to highest participant ID
829 sort idauniq
830 * Drop observations for which the key variable (participant ID) does not match
831 drop if merge_crossweight11==2
832 * Sort from lowest to highest participant ID and wave
833 sort idauniq wave
834
835 * ELSA Covid-19 cross-sectional weight (Core members) - COVID Wave 2
836 * Replace variable as missing for any missing cases (coded as negative numbers in the ELSA dataset)
837 replace wtfin1 = . if wtfin1<0
838 * Tables of frequencies - COVID Wave 2
839 tab activity2 if wave==11
840 tab activity2 if wave==11 [aw=wtfin1]
841 tab shielding if wave==11
842 tab shielding if wave==11 [aw=wtfin1]
843 tab activity2 if wave==11 & wtfin1==.
844 tab shielding if wave==11 & wtfin1==.
845 * Keep data from Wave 9 (baseline) only
846 keep if wave==9
847 * Save dataset with a new name
848 save descw9padatar.dta
849
850 * Tables of frequencies, weighted using the Wave 9 cross-sectional weight
851 tab activity2 [aw=w9xwgt]
852 tab education_cons [aw=w9xwgt]
853 tab mynssec3_cons [aw=w9xwgt]
854 tab wealth_cons [aw=w9xwgt]
855 tab sex_cons [aw=w9xwgt]
856 tab ethnicity_cons [aw=w9xwgt]
857 tab age_cons [aw=w9xwgt]
858 tab living_cons [aw=w9xwgt]
859 tab limiting_cons [aw=w9xwgt]
860 * Summary statistics for continuous variables, weighted using the Wave 9 cross-sectional weight
861 sum indager [aw=w9xwgt]
862 sum depression_cons [aw=w9xwgt]
863
864 * Tables of frequencies, unweighted
865 tab activity2
866 tab education_cons
867 tab mynssec3_cons
868 tab wealth_cons
869 tab sex_cons
870 tab ethnicity_cons
871 tab age_cons
872 tab living_cons
873 tab limiting_cons
874 * Summary statistics for continuous variables, unweighted
875 sum indager
876 sum depression_cons
877
878 * Use dataset with processed variables
879 use data03par.dta
880 * Keep variables required for analyses and multiple imputation
881 keep idauniq wave TimePA edqual nssec8 mynssec3 education education_cons mynssec3_cons wealth_cons
sex_cons ethnicity_cons living_cons age_cons limiting_cons activity2 cov19lwgtw2 health_cons

```

```

depression_cons alcohol_cons smoking_cons shielding
882 * Count total number of participants and observations
883 unique idauniq
884 * 4,407 individuals, 8,814 observations
885
886 * Time-constant education (raw) - Wave 9
887 * Generate a new variable duplicating the raw education variable at Wave 9
888 gen edqual_cons = edqual if wave==9
889 * Declare a panel dataset with participant ID "idauniq" and time variable "wave"
890 tsset idauniq wave
891 * Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID
Wave 2) by participant ID
892 bysort idauniq: carryforward edqual_cons, replace
893
894 * Time-constant occupational class (raw) - Wave 9
895 * Generate a new variable duplicating the raw occupational class variable at Wave 9
896 gen nssec8_cons = nssec8 if wave==9
897 * Declare a panel dataset with participant ID "idauniq" and time variable "wave"
898 tsset idauniq wave
899 * Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID
Wave 2) by participant ID
900 bysort idauniq: carryforward nssec8_cons, replace
901
902 * Assign extended missing value for response "foreign/other" for education
903 replace education_cons = .a if edqual_cons==6
904 * Assign extended missing value for response "never worked and long-term unemployed" for
occupational class
905 replace mynssec3_cons = .a if nssec8_cons==8
906 * Drop unnecessary variables
907 drop nssec8 edqual nssec8_cons edqual_cons mynssec3 education
908
909 * Produce a table with the number of missing values and percent missing for each variable in the list
910 mdesc cov19lwgtw2 activity2 education_cons mynssec3_cons wealth_cons sex_cons ethnicity_cons
living_cons age_cons limiting_cons TimePA health_cons depression_cons alcohol_cons smoking_cons
shielding
911 * Save dataset with a new name
912 save imputationr.dta
913
914 * Multiple imputation
915 * Arrange the multiple datasets in "marginal and long" format
916 mi set mlong
917 * Generate summary of missing values
918 mi misstable summarize activity2 education_cons mynssec3_cons wealth_cons sex_cons ethnicity_cons
living_cons age_cons limiting_cons TimePA health_cons depression_cons alcohol_cons smoking_cons
shielding
919 * Display patterns of missing data
920 mi misstable patterns activity2 education_cons mynssec3_cons wealth_cons sex_cons ethnicity_cons
living_cons age_cons limiting_cons TimePA health_cons depression_cons alcohol_cons smoking_cons
shielding
921 * Generate dummy variables (with prefix miss_ added to each variable name) to be coded 0 if variable
is observed and 1 if the variable has a missing value
922 quietly misstable summarize activity2 education_cons mynssec3_cons wealth_cons sex_cons
ethnicity_cons living_cons age_cons limiting_cons TimePA health_cons depression_cons alcohol_cons
smoking_cons shielding, generate(miss_)
923 * Review changes
924 describe miss_*
925
926 * Ordinal logistic (ologit), logistic (logit), and linear (regress) regression models to explore
whether candidate auxiliary variables predict 1) variables in the analytic models; and 2) missing
data on variables in the analytic models
927 ologit activity2 i.health_cons i.alcohol_cons i.smoking_cons
928 logit miss_activity2 i.health_cons i.alcohol_cons i.smoking_cons
929 ologit education_cons i.health_cons i.alcohol_cons i.smoking_cons

```

```

930  logit miss_education_cons i.health_cons i.alcohol_cons i.smoking_cons if education_cons <= .
931  ologit mynssec3_cons i.health_cons i.alcohol_cons i.smoking_cons
932  logit miss_mynssec3_cons i.health_cons i.alcohol_cons i.smoking_cons if mynssec3_cons <= .
933  ologit wealth_cons i.health_cons i.alcohol_cons i.smoking_cons
934  logit miss_wealth_cons i.health_cons i.alcohol_cons i.smoking_cons
935  logit sex_cons i.health_cons i.alcohol_cons i.smoking_cons
936  logit ethnicity_cons i.health_cons i.alcohol_cons i.smoking_cons
937  logit living_cons i.health_cons i.alcohol_cons i.smoking_cons
938  ologit age_cons i.health_cons i.alcohol_cons i.smoking_cons
939  logit limiting_cons i.health_cons i.alcohol_cons i.smoking_cons
940  logit miss_limiting_cons i.health_cons i.alcohol_cons i.smoking_cons
941  regress depression_cons i.health_cons i.alcohol_cons i.smoking_cons
942  logit miss_depression_cons i.health_cons i.alcohol_cons i.smoking_cons
943  logit shielding i.health_cons i.alcohol_cons i.smoking_cons
944  logit miss_shielding i.health_cons i.alcohol_cons i.smoking_cons
945
946  * Drop unnecessary variables
947  drop miss_* wave
948  * Reshape data into wide format for observations identified by participant ID and add "TimePA" as an
  identifying time period
949  mi reshape wide activity2, i(idauniq) j(TimePA)
950  * Register all variables with missing values that need to be imputed
951  mi register imputed activity20 activity21 education_cons mynssec3_cons wealth_cons limiting_cons
  health_cons depression_cons alcohol_cons shielding
952  * Register all variables with no missing values and/or which do not require imputation
953  mi register regular sex_cons ethnicity_cons living_cons age_cons smoking_cons
954  * Clear panel data settings
955  mi xtset, clear
956
957  * Impute variables
958  * Imputation methods:
959  * ologit: ordinal logistic
960  * logit: logistic
961  * nbreg: negative binomial regression
962  * Notes: The variables on the right of the "=" sign have no missing information and are therefore
  solely considered predictors of missing values. The imputation model is weighted using the Covid-19
  study Wave 2 longitudinal weight. The "add(20)" command specifies the number of imputations to be
  performed; rseed() sets the seed.
963  mi impute chained (ologit) activity20 activity21 education_cons mynssec3_cons wealth_cons health_cons
  alcohol_cons (logit) limiting_cons shielding (nbreg) depression_cons = sex_cons ethnicity_cons
  living_cons age_cons smoking_cons [pweight=cov19lwgtw2], add(20) rseed(54321) noisily force
964  * Save the multiple datasets in wide format
965  save imputedwidepaar.dta
966
967  * Reshape data into long format
968  mi reshape long activity2, i(idauniq) j(TimePA)
969  * Save the multiple datasets in long format
970  save imputedlongpaar.dta
971
972  * GENERALISED LINEAR MIXED MODELS - MULTIPLE IMPUTATION
973  * Use multiply imputed dataset in long format
974  use imputedlongpaar.dta
975  * Display base levels of factor variables and their interactions in output tables
976  set showbaselevels on
977
978  * UNADJUSTED MODELS - MULTIPLE IMPUTATION
979  * meologit: Multilevel mixed-effects ordered logistic regression command
980  * pweight: Incorporates sampling weights at higher levels (i.e., participant level)
981  * or: Reports fixed-effects coefficients as odds ratios
982  * ##: Specifies the main effects for each variable and an interaction
983  * i.: Denotes a factor variable
984  * cmdok: Forces the "meologit" command to run on imputed data
985  * mi estimate: Runs the analytical model (i.e., multilevel ordinal logistic regression) within each

```

```

of the imputed datasets
986 * Model 1: Two-level ordered logit regression of physical activity on indicators for levels of
education and time, and their interaction, with random intercepts by participant ID
987 mi estimate, or cmdok: meologit activity2 i.education_cons##i.TimePA || idauniq:, pweight(cov19lwgtw2)
988 * Model 2: Two-level ordered logit regression of physical activity on indicators for levels of
occupational class and time, and their interaction, with random intercepts by participant ID
989 mi estimate, or cmdok: meologit activity2 i.mynssec3_cons##i.TimePA || idauniq:, pweight(cov19lwgtw2)
990 * Model 3: Two-level ordered logit regression of physical activity on indicators for levels of
wealth and time, and their interaction, with random intercepts by participant ID
991 mi estimate, or cmdok: meologit activity2 i.wealth_cons##i.TimePA || idauniq:, pweight(cov19lwgtw2)
992 * Model 4: Two-level ordered logit regression of physical activity on indicators for levels of
education, occupational class, wealth, and time, including interactions between the three
socio-economic variables and time, with random intercepts by participant ID
993 mi estimate, or cmdok: meologit activity2 i.education_cons##i.TimePA i.mynssec3_cons##i.TimePA i.
wealth_cons##i.TimePA || idauniq:, pweight(cov19lwgtw2)
994
995 * FULLY ADJUSTED MODELS (I.E., WITH COVARIATES) - MULTIPLE IMPUTATION
996 * Model 1: Two-level ordered logit regression of physical activity on indicators for levels of
education and time, and their interaction (adjusted for covariates), with random intercepts by
participant ID
997 mi estimate, or cmdok: meologit activity2 i.education_cons##i.TimePA i.sex_cons i.ethnicity_cons i.
living_cons i.age_cons i.limiting_cons depression_cons i.shielding || idauniq:, pweight(cov19lwgtw2)
998 * Model 2: Two-level ordered logit regression of physical activity on indicators for levels of
occupational class and time, and their interaction (adjusted for covariates), with random intercepts
by participant ID
999 mi estimate, or cmdok: meologit activity2 i.mynssec3_cons##i.TimePA i.sex_cons i.ethnicity_cons i.
living_cons i.age_cons i.limiting_cons depression_cons i.shielding || idauniq:, pweight(cov19lwgtw2)
1000 * Model 3: Two-level ordered logit regression of physical activity on indicators for levels of
wealth and time, and their interaction (adjusted for covariates), with random intercepts by
participant ID
1001 mi estimate, or cmdok: meologit activity2 i.wealth_cons##i.TimePA i.sex_cons i.ethnicity_cons i.
living_cons i.age_cons i.limiting_cons depression_cons i.shielding || idauniq:, pweight(cov19lwgtw2)
1002 * Model 4: Two-level ordered logit regression of physical activity on indicators for levels of
education, occupational class, wealth, and time, including interactions between the three
socio-economic variables and time (adjusted for covariates), with random intercepts by participant ID
1003 mi estimate, or cmdok: meologit activity2 i.education_cons##i.TimePA i.mynssec3_cons##i.TimePA i.
wealth_cons##i.TimePA i.sex_cons i.ethnicity_cons i.living_cons i.age_cons i.limiting_cons
depression_cons i.shielding || idauniq:, pweight(cov19lwgtw2)

```