```
*************************************************************************************************************************
***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"***
*************************************************************************************************************************

* STATA version: 17.0, BE-Basic Edition

* STATA citation: StataCorp. 2021. Stata Statistical Software: Release 17. College Station, TX: StataCorp LLC.

* 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

* 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

* 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).

* Date of data access/download (dd/mm/yyyy): 17/12/2021

* Project ID: 217429

* 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).

**********************
***DATA PROCESSING***
**********************

* Change working directory - add pathname in between quotation marks for Windows
cd ""

* Variables Wave 9
use idauniq heacta heactb heactc w9nssec8 w9nssec3 samptyp w9xwgt w9scwt indsex indager dimarr fqethnmr wpdes hhtot heill helim hehelf 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
* Describe dataset
describe
* Sort from lowest to highest participant identifier (ID)
sort idauniq
* Rename variables to shorter forms
```

```
rename w9nssec8 nssec8
rename w9nssec3 nssec3
rename indsex Sex
* Generate a new variable called wave and assign the number 9 to each observation (to designate Wave 9)
gen wave = 9
* Save Wave 9 core dataset
save wave9panew.dta

* Variables COVID Wave 2
use idauniq HEACTA HEACTB HEACTC Finstat_w1 Cohort CorePartner wtfin1 wtfin2 cov19lwgtw2 cov19lwgtw2b cov19lwgtw2c using elsa_covid_w2_eul.dta
* Describe dataset
describe
* Sort from lowest to highest participant ID
sort idauniq
* Rename variables to shorter forms and to ensure consistency with Wave 9 (for heacta-heactc)
rename HEACTA heacta
rename HEACTB heactb
rename HEACTC heactc
rename Finstat_w1 FinStat
* Generate a new variable called wave and assign the number 11 to each observation (to designate COVID Wave 2)
gen wave = 11
* Save COVID Wave 2 core dataset
save covidwave2panew.dta

* Variables Wave 9 Derived
use idauniq edqual using wave_9_ifs_derived_variables.dta
* Describe dataset
describe
* Sort from lowest to highest participant ID
sort idauniq
* Save Wave 9 derived dataset
save wave9derived.dta

* Variables Wave 9 Financial Derived
use idauniq totwq5_bu_s using wave_9_financial_derived_variables.dta
* Describe dataset
describe
* Sort from lowest to highest participant ID
sort idauniq
* Save Wave 9 financial dataset
save wave9financial.dta

* Wave 9 complete data
* Merge core, derived, and financial datasets for Wave 9 using the participant ID
* Use Wave 9 core dataset
use wave9panew.dta
* One-to-one merge of data in memory with wave9financial.dta on participant ID
merge 1:1 idauniq using wave9financial.dta, generate (merge_financial9)
* Overwrite Wave 9 dataset, by replacing the previously saved file
save wave9panew.dta, replace
* Use the newly saved file for Wave 9
use wave9panew.dta
* One-to-one merge of data in memory with wave9derived.dta on participant ID
merge 1:1 idauniq using wave9derived.dta, generate (merge_derived9)
* Sort from lowest to highest participant ID
sort idauniq
* Overwrite Wave 9 dataset, by replacing the previously saved file
save wave9panew.dta, replace

* Append Wave 9 and COVID Wave 2 datasets
```

```
use wave9panew.dta
append using covidwave2panew.dta
* Sort by participant ID and wave (lowest to highest)
sort idauniq wave
* Assigns a number in ascending order to each row of observations
gen ascnr = _n

* Unique individual serial number (personal ID)
* Replace variable as missing for any missing cases (coded as negative numbers in the ELSA dataset)
replace idauniq = . if idauniq<0

* Organising dataset
* Generate a variable that assigns the observation number (i.e., 1 for first data collection timepoint, 2 for second data collection timepoint) to each row by participant ID
bysort idauniq (wave): gen obsnr = _n
* Generate a variable that assigns the number of total observations to each row of data for a given participant
bysort idauniq: gen obscount = _N
* Check how many participants have data at 1 or 2 timepoints - the "if obsnr==1" statement is used to prevent participants with data at two timepoints from contributing to the counts twice
tabulate obscount if obsnr==1
* Generate a variable that assigns the number 1 to the row representing participants' first observation
bysort idauniq (wave): gen first = 1 if _n==1
* Generate a variable that assigns the number 1 to the row representing participants' last observation
bysort idauniq (wave): gen last = 1 if _n==_N
* Generate a variable that assigns the number 1 to the row representing participants' first observation if this corresponds to Wave 9 (baseline)
bysort idauniq (wave): gen firstwave = 1 if obsnr==1 & wave==9
* Carry the value of this last variable forwards to the remainder of a participant's observations
bysort idauniq: gen variable = firstwave[1]
* Install unique command
ssc install unique
* Count total number of participants and observations
unique idauniq
* 9,014 individuals, 15,530 observations
* Assign the COVID Wave 2 longitudinal weight to all observations for a participant
bysort idauniq(wave): replace cov19lwgtw2 = cov19lwgtw2[2]
* Drop if participant is not a core member (i.e., if they do not have a valid sampling weight assigned)
drop if inlist(cov19lwgtw2,-1,.)
* Count total number of participants and observations
unique idauniq
* 5,378 individuals, 10,756 observations
* Replace age = 90 if participant is aged 90+ years (collapsed in ELSA and coded as -7 at Wave 9)
replace indager = 90 if indager==-7
* Drop observation if the participant is aged less than 60 years at Wave 9
drop if indager < 60 & wave==9
* Count total number of participants and observations
unique idauniq
* 5,378 individuals, 9,785 observations
* Check how many participants have data at Wave 9
tab firstwave
* Drop if age data are missing at Wave 9
drop if indager ==. & wave==9
* Count total number of participants and observations
unique idauniq
* 5,378 individuals, 9,785 observations
* Save dataset with a new name
save datapanew.dta

* Vigorous/Moderate/Mild sports or activities (Wave 9, COVID Wave 2)
* Replace variables as missing for any missing cases (coded as negative numbers in the ELSA dataset)
```

```
replace heacta = . if heacta<0
replace heactb = . if heactb<0
replace heactc = . if heactc<0
* Generate a new variable
gen activity2 = .
* Assign the number 3 if the participant partakes in vigorous activity more than once a week or ("|") once a week
replace activity2 = 3 if heacta==1 | heacta==2
* 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
replace activity2 = 2 if (heactb==1 | heactb==2) & inlist(heacta,3,4)
* 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
replace activity2 = 1 if (heactc==1 | heactc==2) & inlist(heacta,3,4) & inlist(heactb,3,4)
* Assign the number 0 if the participant does not take part in activity of any intensity once a week or more
replace activity2 = 0 if inlist(heacta,3,4) & inlist(heactb,3,4) & inlist(heactc,3,4)
* Replace the variable as missing for participants with missing cases on all three variables
replace activity2 = . if inlist(heacta,.) & inlist(heactb,.) & inlist(heactc,.)
* Coding of final physical activity variable:
* 3: Vigorous activity at least once per week
* 2: At least moderate but no vigorous activity at least once per week
* 1: Only mild activity at least once per week
* 0: Inactive (no activity on a weekly basis)

* Highest Educational Qualification (Wave 9)
* Excluded foreign/other
* Replace variable as missing for any missing cases (coded as negative numbers in the ELSA dataset)
replace edqual = . if edqual<0
* Check participant counts in each category at Wave 9
tab edqual if wave==9
* Generate a new variable
gen education = .
* Assign the number 0 if the participant does not have any formal qualifications
replace education = 0 if edqual == 7
* Assign the number 1 if the participant has A level equivalent, O level equivalent, or other grade equivalent
replace education = 1 if inlist(edqual,3,4,5)
* Assign the number 2 if the participant has completed some higher education (below degree), or has a degree or equivalent
replace education = 2 if inlist(edqual,1,2)
* Coding of final education variable:
* 0: No formal qualifications
* 1: School qualifications
* 2: Higher education

* NS-SEC 8 and 3 category classification (Wave 9)
* Excluded Never worked and long-term unemployed
* Replace variables as missing for any missing cases (coded as negative numbers or 99 in the ELSA dataset)
replace nssec8 = . if nssec8<0
replace nssec8 = . if nssec8 == 99
replace nssec3 = . if nssec3<0
replace nssec3 = . if nssec3 == 99
* Generate a new variable
gen mynssec3 = .
* 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
replace mynssec3 = 2 if inlist(nssec8,1,2)
* 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
replace mynssec3 = 1 if inlist(nssec8,3,4)
```

```
* 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
replace mynssec3 = 0 if inlist(nssec8,5,6,7)
* Coding of final occupational class variable:
* 0: Lower occupations
* 1: Intermediate occupations
* 2: Higher occupations

* Quintiles of BU total (non-pension) wealth (Wave 9)
* Replace variable as missing for any missing cases (coded as negative numbers in the ELSA dataset)
replace totwq5_bu_s = . if totwq5_bu_s<0
* Coding of final wealth variable:
* 1: 1st quintile (lowest)
* 2: 2nd quintile
* 3: 3rd quintile
* 4: 4th quintile
* 5: 5th quintile (highest)

* Wave 9 cross-sectional weight
* Replace variable as missing for any missing cases (coded as negative numbers in the ELSA dataset)
replace w9xwgt = . if w9xwgt<0

* ELSA Covid-19 cross-sectional weight (Core members) - COVID Wave 2
* Replace variable as missing for any missing cases (coded as negative numbers in the ELSA dataset)
replace wtfin1 = . if wtfin1<0

* ELSA Covid-19 study Wave 2 longitudinal weight (covid w2 vs ELSA w9)
* Replace variable as missing for any missing cases (coded as negative numbers in the ELSA dataset)
replace cov19lwgtw2 = . if cov19lwgtw2<0

* Biological sex (Wave 9)
* Replace variable as missing for any missing cases (coded as negative numbers in the ELSA dataset)
replace Sex = . if Sex<0
* Assign the number 0 if the participant is male
replace Sex = 0 if Sex == 1
* Assign the number 1 if the participant is female
replace Sex = 1 if Sex == 2
* Coding of the final biological sex variable:
* 0: Male, 1: Female

* Current legal marital status (Wave 9)
* Replace variable as missing for any missing cases (coded as negative numbers in the ELSA dataset)
replace dimarr = . if dimarr<0
* Check participant counts in each category at Wave 9
tab dimarr
* Generate a new variable
gen marital = .
* Assign the number 0 if the participant's marital status was coded as: Single, that is never married and never registered in a same-sex Civil Partnership
replace marital = 0 if dimarr == 1
* Assign the number 1 if the participant's marital status was coded as: Separated, but still legally married or (spontaneous only) in a same-sex Civil Partnership; or Divorced or (spontaneous only) formerly in a same-sex Civil Partnership; or Widowed or (spontaneous only) a surviving civil partner from a same-sex Civil Partnership
replace marital = 1 if inlist(dimarr,4,5,6)
* Assign the number 2 if the participant's marital status was coded as: Married, first and only marriage or a civil partner in a registered same-sex Civil Partnership; or Remarried, second or later marriage
replace marital = 2 if inlist(dimarr,2,3)
* Coding of the final marital status variable:
* 0: Single/Never married/Never registered in a Civil Partnership
* 1: Separated/Divorced/Widowed
* 2: Married/Remarried/In a registered Civil Partnership
```

```
261
262 * Ethnicity (Wave 9)
263 * Replace variable as missing for any missing cases (coded as negative numbers in the ELSA dataset)
264 replace fqethnmr = . if fqethnmr<0
265 * Assign the number 0 if the participant is White
266 replace fqethnmr = 0 if fqethnmr == 1
267 * Assign the number 1 if the participant is Non-White
268 replace fqethnmr = 1 if fqethnmr == 2
269 * Coding of the final ethnicity variable:
270 * 0: White, 1: Non-White
271
272 * Current employment situation (Wave 9)
273 * Replace variable as missing for any missing cases (coded as negative numbers in the ELSA dataset)
274 replace wpdes = . if wpdes<0
275 * Generate a new variable
276 gen employment = .
277 * Assign the number 0 if the participant's employment status was coded as: Retired; or Unemployed;
    or Permanently sick or disabled; or Looking after home or family
278 replace employment = 0 if inlist(wpdes,1,4,5,6)
279 * Assign the number 1 if the participant's employment status was coded as: Employed; or
    Self-employed; or SPONTANEOUS: Semi-retired
280 replace employment = 1 if inlist(wpdes,2,3,96)
281 * Coding of the final employment status variable:
282 * 0: Not working, 1: Working full- or part-time
283
284 * Number of people in household (Wave 9)
285 * Replace variable as missing for any missing cases (coded as negative numbers or 0 in the ELSA
    dataset)
286 replace hhtot = . if hhtot<0
287 replace hhtot = . if hhtot==0
288 * Assign the number 0 if one person lives in household
289 replace hhtot = 0 if hhtot==1
290 * Assign the number 1 if more than one person lives in household
291 replace hhtot = 1 if hhtot>1 & hhtot != .
292 * Coding of the final living status variable:
293 * 0: Living alone, 1: Not living alone
294
295 * Age categorical (Wave 9)
296 * Generate a new variable
297 gen age_cat = .
298 * Assign the number 0 for participants aged 60-69 years at Wave 9
299 replace age_cat = 0 if indager >= 60 & indager <= 69
300 * Assign the number 1 for participants aged 70-79 years at Wave 9
301 replace age_cat = 1 if indager >= 70 & indager <= 79
302 * Assign the number 2 for participants aged 80+ years at Wave 9 and without missing age data
303 replace age_cat = 2 if indager >= 80 & indager != .
304 * Coding of the final categorical age variable:
305 * 0: 60-69 years
306 * 1: 70-79 years
307 * 2: 80+ years
308
309 * Limiting long-standing illness (Wave 9)
310 * 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
311 gen limiting = 0 if heill == 2 | helim == 2
312 * Assign the number 1 for participants with a limiting long-standing illness
313 replace limiting = 1 if helim == 1
314 * Coding of the final limiting long-standing illness variable:
315 * 0: No long-standing illness or not limiting, 1: Limiting long-standing illness
316
317 * Save dataset with a new name
318 save data01panew.dta
319
```

```
* Time-constant education - Wave 9
* Generate a new variable duplicating the education variable at Wave 9
gen education_cons = education if wave==9
* Declare a panel dataset with participant ID "idauniq" and time variable "wave"
tsset idauniq wave
* Install carryforward command
ssc install carryforward
* Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID Wave 2) by participant ID
bysort idauniq: carryforward education_cons, replace

* Time-constant occupational class - Wave 9
* Generate a new variable duplicating the occupational class variable at Wave 9
gen mynssec3_cons = mynssec3 if wave==9
* Declare a panel dataset with participant ID "idauniq" and time variable "wave"
tsset idauniq wave
* Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID Wave 2) by participant ID
bysort idauniq: carryforward mynssec3_cons, replace

* Time-constant wealth - Wave 9
* Generate a new variable duplicating the wealth variable at Wave 9
gen wealth_cons = totwq5_bu_s if wave==9
* Declare a panel dataset with participant ID "idauniq" and time variable "wave"
tsset idauniq wave
* Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID Wave 2) by participant ID
bysort idauniq: carryforward wealth_cons, replace

* Time-constant biological sex - Wave 9
* Generate a new variable duplicating the biological sex variable at Wave 9
gen sex_cons = Sex if wave==9
* Declare a panel dataset with participant ID "idauniq" and time variable "wave"
tsset idauniq wave
* Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID Wave 2) by participant ID
bysort idauniq: carryforward sex_cons, replace

* Time-constant marital status - Wave 9
* Generate a new variable duplicating the marital status variable at Wave 9
gen marital_cons = marital if wave==9
* Declare a panel dataset with participant ID "idauniq" and time variable "wave"
tsset idauniq wave
* Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID Wave 2) by participant ID
bysort idauniq: carryforward marital_cons, replace

* Time-constant ethnicity - Wave 9
* Generate a new variable duplicating the ethnicity variable at Wave 9
gen ethnicity_cons = fqethnmr if wave==9
* Declare a panel dataset with participant ID "idauniq" and time variable "wave"
tsset idauniq wave
* Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID Wave 2) by participant ID
bysort idauniq: carryforward ethnicity_cons, replace

* Time-constant employment - Wave 9
* Generate a new variable duplicating the employment status variable at Wave 9
gen employment_cons = employment if wave==9
* Declare a panel dataset with participant ID "idauniq" and time variable "wave"
tsset idauniq wave
* Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID Wave 2) by participant ID
```

```
bysort idauniq: carryforward employment_cons, replace

* Time-constant living status - Wave 9
* Generate a new variable duplicating the living status variable at Wave 9
gen living_cons = hhtot if wave==9
* Declare a panel dataset with participant ID "idauniq" and time variable "wave"
tsset idauniq wave
* Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID Wave 2) by participant ID
bysort idauniq: carryforward living_cons, replace

* Time-constant age category - Wave 9
* Generate a new variable duplicating the categorical age variable at Wave 9
gen age_cons = age_cat if wave==9
* Declare a panel dataset with participant ID "idauniq" and time variable "wave"
tsset idauniq wave
* Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID Wave 2) by participant ID
bysort idauniq: carryforward age_cons, replace

* Time-constant limiting long-standing illness - Wave 9
* Generate a new variable duplicating the limiting long-standing illness variable at Wave 9
gen limiting_cons = limiting if wave==9
* Declare a panel dataset with participant ID "idauniq" and time variable "wave"
tsset idauniq wave
* Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID Wave 2) by participant ID
bysort idauniq: carryforward limiting_cons, replace

* Time variable
* Generate a new variable
gen TimePA = .
* Assign the number 0 for observations at Wave 9
replace TimePA = 0 if wave==9
* Assign the number 1 for observations at COVID Wave 2
replace TimePA = 1 if wave==11
* Coding of the final time variable:
* 0: Wave 9, 1: COVID Wave 2

* Self-rated health (Wave 9)
* Generate a new variable duplicating the self-rated health variable
gen health = hehelf
* Replace variable as missing for any missing cases (coded as negative numbers in the ELSA dataset)
replace health = . if health<0
* Reverse the self-rated health variable (this creates a new variable and adds the "rev" prefix to the original variable name)
revrs health
* Generate a new variable duplicating the reversed (revhealth) self-rated health variable at Wave 9
gen health_cons = revhealth if wave==9
* Declare a panel dataset with participant ID "idauniq" and time variable "wave"
tsset idauniq wave
* Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID Wave 2) by participant ID
bysort idauniq: carryforward health_cons, replace
* Coding of the final self-rated health variable:
* 1: Poor
* 2: Fair
* 3: Good
* 4: Very good
* 5: Excellent

* Depressive symptoms (Wave 9)
* Recode to the number 0 if participant answered "No" (items psceda-pscedc are reverse-coded)
```

```
replace psceda = 0 if psceda==2
replace pscedb = 0 if pscedb==2
replace pscedc = 0 if pscedc==2

* Recode to the number 0 if participant answered "Yes"
replace pscedd = 0 if pscedd==1
* Recode to the number 1 if participant answered "No"
replace pscedd = 1 if pscedd==2

* Recode to the number 0 if participant answered "No" (item pscede is reverse-coded)
replace pscede = 0 if pscede==2

* Recode to the number 0 if participant answered "Yes"
replace pscedf = 0 if pscedf==1
* Recode to the number 1 if participant answered "No"
replace pscedf = 1 if pscedf==2

* Recode to the number 0 if participant answered "No" (items pscedg-pscedh are reverse-coded)
replace pscedg = 0 if pscedg==2
replace pscedh = 0 if pscedh==2

* Generate new variables duplicating psceda-pscedh, but excluding missing cases (coded as negative
numbers in the ELSA dataset)
gen ceda = psceda if psceda>=0
gen cedb = pscedb if pscedb>=0
gen cedc = pscedc if pscedc>=0
gen cedd = pscedd if pscedd>=0
gen cede = pscede if pscede>=0
gen cedf = pscedf if pscedf>=0
gen cedg = pscedg if pscedg>=0
gen cedh = pscedh if pscedh>=0

* Generate a new variable equal to the sum of depressive symptoms (eight items) to create a total
depression score
gen depression = ceda + cedb + cedc + cedd + cede + cedf + cedg + cedh
* Generate a new variable duplicating the depressive symptoms variable at Wave 9
gen depression_cons = depression if wave==9
* Declare a panel dataset with participant ID "idauniq" and time variable "wave"
tsset idauniq wave
* Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID
Wave 2) by participant ID
bysort idauniq: carryforward depression_cons, replace

* Alcohol consumption (Wave 9)
* 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
gen alcohol = 0 if inlist(scalcm,5,6,7,8)
* 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
replace alcohol = 1 if inlist(scalcm,3,4)
* Assign the number 2 if the participant's alcohol consumption was coded as: Almost every day; or
Five or six days a week
replace alcohol = 2 if inlist(scalcm,1,2)
* Generate a new variable duplicating the alcohol consumption variable at Wave 9
gen alcohol_cons = alcohol if wave==9
* Declare a panel dataset with participant ID "idauniq" and time variable "wave"
tsset idauniq wave
* Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID
Wave 2) by participant ID
bysort idauniq: carryforward alcohol_cons, replace
* Coding of the final alcohol consumption variable:
* 0: Less than once a week
```

```
* 1: One to four times per week
* 2: Five or more times per week

* Smoking status (Wave 9)
* Generate a new variable and assign the number 0 for participants who do not smoke cigarettes at all nowadays
gen smoking = 0 if heska==2
* Assign the number 0 for participants who never smoked cigarettes
replace smoking = 0 if hesmk==2
* Assign the number 1 for participants who smoke cigarettes nowadays (heska)
replace smoking = 1 if heska==1
* Assign the number 1 for participants who do smoke cigarettes nowadays (heskf)
replace smoking = 1 if heskf==1
* * Generate a new variable duplicating the smoking status variable at Wave 9
gen smoking_cons = smoking if wave==9
* Declare a panel dataset with participant ID "idauniq" and time variable "wave"
tsset idauniq wave
* Carryforward observations with respect to the time variable "wave" (i.e., from Wave 9 to COVID Wave 2) by participant ID
bysort idauniq: carryforward smoking_cons, replace
* Coding of the final smoking status variable:
* 0: Not currently smoking, 1: Current smoker

* Save dataset with a new name
save data02panew.dta

* Keep variables required for analyses
keep idauniq wave TimePA wpdes education_cons mynssec3_cons wealth_cons sex_cons marital_cons ethnicity_cons employment_cons living_cons age_cons limiting_cons activity2 cov19lwgtw2
* Count total number of participants and observations
unique idauniq
* 5,378 individuals, 9,785 observations
* Generate a variable that assigns the number of total observations to each row of data for a given participant
bysort idauniq: gen obscount = _N
* Keep participants that have at least some data at both timepoints of interest (i.e., Wave 9 and COVID Wave 2)
keep if obscount==2
* Drop unnecessary variable
drop obscount
* Count total number of participants and observations
unique idauniq
* 4,407 individuals, 8,814 observations
* Produce a table with the number of missing values and percent missing for each variable in the list
mdesc cov19lwgtw2 activity education_cons mynssec3_cons wealth_cons sex_cons marital_cons ethnicity_cons employment_cons living_cons age_cons limiting_cons TimePA

* Drop observation if physical activity data are missing
drop if activity2 == .
* Generate a variable that assigns the number of total observations to each row of data for a given participant
bysort idauniq: gen obscount = _N
* Keep participants that have at least some data at both timepoints of interest (i.e., Wave 9 and COVID Wave 2)
keep if obscount==2
* Drop unnecessary variable
drop obscount
* Count total number of participants and observations
unique idauniq
* 4,404 individuals, 8,808 observations
* Drop observation if education, occupational class, or wealth data are missing
drop if education_cons == . | mynssec3_cons == . | wealth_cons == .
* Generate a variable that assigns the number of total observations to each row of data for a given
```

```
participant
bysort idauniq: gen obscount = _N
* Keep participants that have at least some data at both timepoints of interest (i.e., Wave 9 and COVID Wave 2)
keep if obscount==2
* Drop unnecessary variable
drop obscount
* Count total number of participants and observations
unique idauniq
* 3,802 individuals, 7,604 observations
* Drop if biological sex, marital status, ethnicity, employment status, living status, categorical age, or limiting long-standing illness data are missing
drop if sex_cons == . | marital_cons == . | ethnicity_cons == . | employment_cons == . | living_cons == . | age_cons == . | limiting_cons == .
* Generate a variable that assigns the number of total observations to each row of data for a given participant
bysort idauniq: gen obscount = _N
* Keep participants that have at least some data at both timepoints of interest (i.e., Wave 9 and COVID Wave 2)
keep if obscount==2
* Drop unnecessary variable
drop obscount
* Count total number of participants and observations
unique idauniq
* 3,791 individuals, 7,582 observations

* Save complete case dataset with a new name
save CCpaweightnotcorepartnernew.dta

*************************
***STATISTICAL ANALYSES***
*************************

* GENERALISED LINEAR MIXED MODELS
* Use complete case dataset
use CCpaweightnotcorepartnernew.dta
* Display base levels of factor variables and their interactions in output tables
set showbaselevels on

* UNADJUSTED MODELS
* meologit: Multilevel mixed-effects ordered logistic regression command
* pweight: Incorporates sampling weights at higher levels (i.e., participant level)
* or: Reports fixed-effects coefficients as odds ratios
* ##: Specifies the main effects for each variable and an interaction
* i.: Denotes a factor variable
* 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
meologit activity2 i.education_cons##i.TimePA || idauniq:, pweight(cov19lwgtw2) or
* 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
meologit activity2 i.mynssec3_cons##i.TimePA || idauniq:, pweight(cov19lwgtw2) or
* 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
meologit activity2 i.wealth_cons##i.TimePA || idauniq:, pweight(cov19lwgtw2) or
* 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
meologit activity2 i.education_cons##i.TimePA i.mynssec3_cons##i.TimePA i.wealth_cons##i.TimePA || idauniq:, pweight(cov19lwgtw2) or
* Model 5: Two-level ordered logit regression of physical activity on indicators for levels of education, biological sex, and time, including two-way (between education and time, and between education and biological sex) and three-way (between education, biological sex, and time) interactions, with random intercepts by participant ID
```

```
592  meologit activity2 i.education_cons##i.TimePA i.education_cons##i.sex_cons i.education_cons#i.TimePA#
     i.sex_cons || idauniq:, pweight(cov19lwgtw2) or
593  * Model 6: Two-level ordered logit regression of physical activity on indicators for levels of
     occupational class, biological sex, and time, including two-way (between occupational class and
     time, and between occupational class and biological sex) and three-way (between occupational class,
     biological sex, and time) interactions, with random intercepts by participant ID
594  meologit activity2 i.mynssec3_cons##i.TimePA i.mynssec3_cons##i.sex_cons i.mynssec3_cons#i.TimePA#i.
     sex_cons || idauniq:, pweight(cov19lwgtw2) or
595  * Model 7: Two-level ordered logit regression of physical activity on indicators for levels of
     wealth, biological sex, and time, including two-way (between wealth and time, and between wealth and
     biological sex) and three-way (between wealth, biological sex, and time) interactions, with random
     intercepts by participant ID
596  meologit activity2 i.wealth_cons##i.TimePA i.wealth_cons##i.sex_cons i.wealth_cons#i.TimePA#i.
     sex_cons || idauniq:, pweight(cov19lwgtw2) or
597
598  * FULLY ADJUSTED MODELS (I.E., WITH COVARIATES)
599  * 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
600  meologit activity2 i.education_cons##i.TimePA i.sex_cons i.marital_cons i.ethnicity_cons i.
     employment_cons i.living_cons i.age_cons i.limiting_cons || idauniq:, pweight(cov19lwgtw2) or
601  * vce(unconditional): produces standard errors that account for the sampling variability of
     covariates arising with complex survey data
602  * 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
603  margins i.education_cons i.TimePA i.education_cons#i.TimePA, predict (mu fixedonly outcome(0))
     vsquish vce(unconditional)
604  * 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
605  margins i.education_cons i.TimePA i.education_cons#i.TimePA, predict (mu fixedonly outcome(1))
     vsquish vce(unconditional)
606  * 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
607  margins i.education_cons i.TimePA i.education_cons#i.TimePA, predict (mu fixedonly outcome(2))
     vsquish vce(unconditional)
608  * 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
609  margins i.education_cons i.TimePA i.education_cons#i.TimePA, predict (mu fixedonly outcome(3))
     vsquish vce(unconditional)
610  * 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
611  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)
612
613  * 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
614  meologit activity2 i.mynssec3_cons##i.TimePA i.sex_cons i.marital_cons i.ethnicity_cons i.
     employment_cons i.living_cons i.age_cons i.limiting_cons || idauniq:, pweight(cov19lwgtw2) or
615  * vce(unconditional): produces standard errors that account for the sampling variability of
     covariates arising with complex survey data
616  * 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
617  margins i.mynssec3_cons i.TimePA i.mynssec3_cons#i.TimePA, predict (mu fixedonly outcome(0)) vsquish
     vce(unconditional)
618  * 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
margins i.mynssec3_cons i.TimePA i.mynssec3_cons#i.TimePA, predict (mu fixedonly outcome(1)) vsquish vce(unconditional)
* 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
margins i.mynssec3_cons i.TimePA i.mynssec3_cons#i.TimePA, predict (mu fixedonly outcome(2)) vsquish vce(unconditional)
* 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
margins i.mynssec3_cons i.TimePA i.mynssec3_cons#i.TimePA, predict (mu fixedonly outcome(3)) vsquish vce(unconditional)
* 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
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)

* 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
meologit activity2 i.wealth_cons##i.TimePA i.sex_cons i.marital_cons i.ethnicity_cons i.employment_cons i.living_cons i.age_cons i.limiting_cons || idauniq:, pweight(cov19lwgtw2) or
* vce(unconditional): produces standard errors that account for the sampling variability of covariates arising with complex survey data
* 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
margins i.wealth_cons i.TimePA i.wealth_cons#i.TimePA, predict (mu fixedonly outcome(0)) vsquish vce(unconditional)
* 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
margins i.wealth_cons i.TimePA i.wealth_cons#i.TimePA, predict (mu fixedonly outcome(1)) vsquish vce(unconditional)
* 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
margins i.wealth_cons i.TimePA i.wealth_cons#i.TimePA, predict (mu fixedonly outcome(2)) vsquish vce(unconditional)
* 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
margins i.wealth_cons i.TimePA i.wealth_cons#i.TimePA, predict (mu fixedonly outcome(3)) vsquish vce(unconditional)
* 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
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)

* 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
meologit activity2 i.education_cons##i.TimePA i.mynssec3_cons##i.TimePA i.wealth_cons##i.TimePA i.sex_cons i.marital_cons i.ethnicity_cons i.employment_cons i.living_cons i.age_cons i.limiting_cons || idauniq:, pweight(cov19lwgtw2) or
* vce(unconditional): produces standard errors that account for the sampling variability of covariates arising with complex survey data
* 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
```

```
645  margins i.education_cons i.TimePA i.education_cons#i.TimePA, predict (mu fixedonly outcome(0))
     vsquish vce(unconditional)
646  * 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
647  margins i.education_cons i.TimePA i.education_cons#i.TimePA, predict (mu fixedonly outcome(1))
     vsquish vce(unconditional)
648  * 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
649  margins i.education_cons i.TimePA i.education_cons#i.TimePA, predict (mu fixedonly outcome(2))
     vsquish vce(unconditional)
650  * 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
651  margins i.education_cons i.TimePA i.education_cons#i.TimePA, predict (mu fixedonly outcome(3))
     vsquish vce(unconditional)
652  * 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
653  margins i.mynssec3_cons i.TimePA i.mynssec3_cons#i.TimePA, predict (mu fixedonly outcome(0)) vsquish
     vce(unconditional)
654  * 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
655  margins i.mynssec3_cons i.TimePA i.mynssec3_cons#i.TimePA, predict (mu fixedonly outcome(1)) vsquish
     vce(unconditional)
656  * 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
657  margins i.mynssec3_cons i.TimePA i.mynssec3_cons#i.TimePA, predict (mu fixedonly outcome(2)) vsquish
     vce(unconditional)
658  * 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
659  margins i.mynssec3_cons i.TimePA i.mynssec3_cons#i.TimePA, predict (mu fixedonly outcome(3)) vsquish
     vce(unconditional)
660  * 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
661  margins i.wealth_cons i.TimePA i.wealth_cons#i.TimePA, predict (mu fixedonly outcome(0)) vsquish vce(
     unconditional)
662  * 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
663  margins i.wealth_cons i.TimePA i.wealth_cons#i.TimePA, predict (mu fixedonly outcome(1)) vsquish vce(
     unconditional)
664  * 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
665  margins i.wealth_cons i.TimePA i.wealth_cons#i.TimePA, predict (mu fixedonly outcome(2)) vsquish vce(
     unconditional)
666  * 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
667  margins i.wealth_cons i.TimePA i.wealth_cons#i.TimePA, predict (mu fixedonly outcome(3)) vsquish vce(
     unconditional)
668  * 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
669  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)
670  * 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
671 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)
672 * 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
673 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)
674
675 * Model 5: Two-level ordered logit regression of physical activity on indicators for levels of education, biological sex, and time, including two-way (between education and time, and between education and biological sex) and three-way (between education, biological sex, and time) interactions (adjusted for covariates), with random intercepts by participant ID
676 meologit activity2 i.education_cons##i.TimePA i.education_cons##i.sex_cons i.education_cons#i.TimePA#i.sex_cons i.marital_cons i.ethnicity_cons i.employment_cons i.living_cons i.age_cons i.limiting_cons || idauniq:, pweight(cov19lwgtw2) or
677 * Model 6: Two-level ordered logit regression of physical activity on indicators for levels of occupational class, biological sex, and time, including two-way (between occupational class and time, and between occupational class and biological sex) and three-way (between occupational class, biological sex, and time) interactions (adjusted for covariates), with random intercepts by participant ID
678 meologit activity2 i.mynssec3_cons##i.TimePA i.mynssec3_cons##i.sex_cons i.mynssec3_cons#i.TimePA#i.sex_cons i.marital_cons i.ethnicity_cons i.employment_cons i.living_cons i.age_cons i.limiting_cons || idauniq:, pweight(cov19lwgtw2) or
679 * Model 7: Two-level ordered logit regression of physical activity on indicators for levels of wealth, biological sex, and time, including two-way (between wealth and time, and between wealth and biological sex) and three-way (between wealth, biological sex, and time) interactions (adjusted for covariates), with random intercepts by participant ID
680 meologit activity2 i.wealth_cons##i.TimePA i.wealth_cons##i.sex_cons i.wealth_cons#i.TimePA#i.sex_cons i.marital_cons i.ethnicity_cons i.employment_cons i.living_cons i.age_cons i.limiting_cons || idauniq:, pweight(cov19lwgtw2) or
681
682 * Descriptive statistics
683 * Use participant ID and cross-sectional weight from Wave 9 core dataset
684 use idauniq w9xwgt using wave_9_elsa_data_eul_v1.dta
685 * Describe dataset
686 describe
687 * Sort from lowest to highest participant ID
688 sort idauniq
689 * Generate a new variable called wave and assign the number 9 to each observation (to designate Wave 9)
690 gen wave=9
691 * Save dataset with a new name
692 save wave9crossweight.dta
693
694 * Use complete case dataset
695 use CCpaweightnotcorepartnernew.dta
696 * One-to-one merge of data in memory with wave9crossweight.dta on participant ID and wave
697 merge 1:1 idauniq wave using wave9crossweight.dta, generate(merge_crossweight9)
698 * Sort from lowest to highest participant ID
699 sort idauniq
700 * Drop observations for which the key variable (participant ID) does not match
701 drop if merge_crossweight9==2
702 * Sort from lowest to highest participant ID and wave
703 sort idauniq wave
704 * Assign the Wave 9 cross-sectional weight to all observations for a participant
705 bysort idauniq(wave): replace w9xwgt = w9xwgt[1]
706 * Keep data from Wave 9 (baseline) only
707 keep if wave==9
708 * Save dataset with a new name
709 save dataPADESC.dta
710
```

```
* Tables of frequencies for education, occupational class, wealth, biological sex, ethnicity, and categorical age, weighted using the Wave 9 cross-sectional weight
tab education_cons [aw=w9xwgt]
tab mynssec3_cons [aw=w9xwgt]
tab wealth_cons [aw=w9xwgt]
tab sex_cons [aw=w9xwgt]
tab ethnicity_cons [aw=w9xwgt]
tab age_cons [aw=w9xwgt]

* Use participant ID, marital status, and employment status from Wave 9 core dataset
use idauniq dimarr wpdes using wave_9_elsa_data_eul_v1.dta
* Describe dataset
describe
* Sort from lowest to highest participant ID
sort idauniq
* Generate a new variable called wave and assign the number 9 to each observation (to designate Wave 9)
gen wave=9
* Save dataset with a new name
save wave9desc.dta

* Use participant ID and (continuous) age from Wave 9 core dataset
use idauniq indager using wave_9_elsa_data_eul_v1.dta
* Describe dataset
describe
* Sort from lowest to highest participant ID
sort idauniq
* Generate a new variable called wave and assign the number 9 to each observation (to designate Wave 9)
gen wave=9
* Save dataset with a new name
save wave9indager.dta

* Use Wave 9 dataset for the complete case analytical sample containing information on descriptive variables
use dataPADESC.dta
* One-to-one merge of data in memory with wave9desc.dta on participant ID and wave
merge 1:1 idauniq wave using wave9desc.dta, generate (merge_desc9)
* Drop observations for which the key variable (participant ID) does not match
drop if merge_desc9==2
* Sort from lowest to highest participant ID and wave
sort idauniq wave
* One-to-one merge of data in memory with wave9derived.dta on participant ID and wave
merge 1:1 idauniq using wave9derived.dta, generate (merge_derived9)
* Drop observations for which the key variable (participant ID) does not match
drop if merge_derived9==2
* Sort from lowest to highest participant ID and wave
sort idauniq wave
* One-to-one merge of data in memory with wave9indager.dta on participant ID and wave
merge 1:1 idauniq wave using wave9indager.dta, generate (merge_indager9)
* Drop observations for which the key variable (participant ID) does not match
drop if merge_indager9==2
* Sort from lowest to highest participant ID and wave
sort idauniq wave
* Overwrite dataset, by replacing the previously saved file
save dataPADESC.dta, replace

* Tables of frequencies for marital status, employment status, living status, education (expanded), physical activity, and limiting long-standing illness, weighted using the Wave 9 cross-sectional weight
tab dimarr [aw=w9xwgt]
tab wpdes [aw=w9xwgt]
tab living_cons [aw=w9xwgt]
```

```
tab edqual [aw=w9xwgt]
tab activity [aw=w9xwgt]
tab limiting_cons [aw=w9xwgt]
* Replace age = 90 if participant is aged 90+ years (collapsed in ELSA and coded as -7 at Wave 9)
replace indager = 90 if indager==-7
* Summary statistics for (continuous) age, weighted using the Wave 9 cross-sectional weight
sum indager [aw=w9xwgt]
* Overwrite dataset, by replacing the previously saved file
save dataPADESC.dta, replace

* Use dataset with processed variables
use data02panew.dta
* Keep variables required for analyses and multiple imputation
keep idauniq wave TimePA wpdes education_cons mynssec3_cons wealth_cons sex_cons marital_cons
ethnicity_cons employment_cons living_cons age_cons limiting_cons activity2 cov19lwgtw2 health_cons
depression_cons alcohol_cons smoking_cons
* Count total number of participants and observations
unique idauniq
* 5,378 individuals, 9,785 observations
* Generate a variable that assigns the number of total observations to each row of data for a given
participant
bysort idauniq: gen obscount = _N
* Keep participants that have at least some data at both timepoints of interest (i.e., Wave 9 and
COVID Wave 2)
keep if obscount==2
* Drop unnecessary variable
drop obscount
* Count total number of participants and observations
unique idauniq
* 4,407 individuals, 8,814 observations

* Save dataset with a new name
save toimpute.dta

* Produce a table with the number of missing values and percent missing for each variable in the list
mdesc cov19lwgtw2 activity2 education_cons mynssec3_cons wealth_cons sex_cons marital_cons
ethnicity_cons employment_cons living_cons age_cons limiting_cons TimePA health_cons depression_cons
alcohol_cons smoking_cons
* Drop unnecessary variable
drop wpdes
* Overwrite dataset, by replacing the previously saved file
save toimpute.dta, replace

* Multiple imputation
* Arrange the multiple datasets in "marginal and long" format
mi set mlong
* Generate summary of missing values
mi misstable summarize activity2 education_cons mynssec3_cons wealth_cons sex_cons marital_cons
ethnicity_cons employment_cons living_cons age_cons limiting_cons TimePA health_cons depression_cons
alcohol_cons smoking_cons
* Display patterns of missing data
mi misstable patterns activity2 education_cons mynssec3_cons wealth_cons sex_cons marital_cons
ethnicity_cons employment_cons living_cons age_cons limiting_cons TimePA health_cons depression_cons
alcohol_cons smoking_cons
* 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
quietly misstable summarize activity2 education_cons mynssec3_cons wealth_cons sex_cons marital_cons
ethnicity_cons employment_cons living_cons age_cons limiting_cons TimePA health_cons depression_cons
alcohol_cons smoking_cons, generate(miss_)
* Review changes
describe miss_*

* Ordinal logistic (ologit), multinomial logistic (mlogit), and logistic (logit) 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
818 ologit activity2 i.health_cons depression_cons i.alcohol_cons i.smoking_cons
819 logit miss_activity2 i.health_cons depression_cons i.alcohol_cons i.smoking_cons
820 ologit education_cons i.health_cons depression_cons i.alcohol_cons i.smoking_cons
821 logit miss_education_cons i.health_cons depression_cons i.alcohol_cons i.smoking_cons
822 ologit mynssec3_cons i.health_cons depression_cons i.alcohol_cons i.smoking_cons
823 logit miss_mynssec3_cons i.health_cons depression_cons i.alcohol_cons i.smoking_cons
824 ologit wealth_cons i.health_cons depression_cons i.alcohol_cons i.smoking_cons
825 logit miss_wealth_cons i.health_cons depression_cons i.alcohol_cons i.smoking_cons
826 mlogit marital_cons i.health_cons depression_cons i.alcohol_cons i.smoking_cons
827 logit miss_marital_cons i.health_cons depression_cons i.alcohol_cons i.smoking_cons
828 logit employment_cons i.health_cons depression_cons i.alcohol_cons i.smoking_cons
829 logit miss_employment_cons i.health_cons depression_cons i.alcohol_cons i.smoking_cons
830 logit limiting_cons i.health_cons depression_cons i.alcohol_cons i.smoking_cons
831 logit miss_limiting_cons i.health_cons depression_cons i.alcohol_cons i.smoking_cons
832
833 * Drop unnecessary variables
834 drop miss_* wave
835 * Reshape data into wide format for observations identified by participant ID and add "TimePA" as an
    identifying time period
836 mi reshape wide activity2, i(idauniq) j(TimePA)
837 * Register all variables with missing values that need to be imputed
838 mi register imputed activity20 activity21 education_cons mynssec3_cons wealth_cons marital_cons
    employment_cons limiting_cons health_cons depression_cons alcohol_cons
839 * Register all variables with no missing values and/or which do not require imputation
840 mi register regular sex_cons ethnicity_cons living_cons age_cons smoking_cons
841 * Clear panel data settings
842 mi xtset, clear
843
844 * Impute variables
845 * Imputation methods:
846 * ologit: ordinal logistic
847 * mlogit: multinomial logistic
848 * logit: logistic
849 * nbreg: negative binomial regression
850 * 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. The imputation model was stratified by biological sex.
851 mi impute chained (ologit) activity20 activity21 education_cons mynssec3_cons wealth_cons health_cons
     alcohol_cons (mlogit) marital_cons (logit) employment_cons limiting_cons (nbreg) depression_cons =
    ethnicity_cons living_cons age_cons smoking_cons [pweight=cov19lwgtw2], add(20) rseed(54321) by(
    sex_cons) noisily
852 * Save the multiple datasets in wide format
853 save toimputewidepa.dta
854
855 * Reshape data into long format
856 mi reshape long activity2, i(idauniq) j(TimePA)
857 * Save the multiple datasets in long format
858 save toimputelongpa.dta
859
860 * GENERALISED LINEAR MIXED MODELS - MULTIPLE IMPUTATION
861 * Use multiply imputed dataset in long format
862 use toimputelongpa.dta
863 * Display base levels of factor variables and their interactions in output tables
864 set showbaselevels on
865
866 * UNADJUSTED MODELS - MULTIPLE IMPUTATION
867 * meologit: Multilevel mixed-effects ordered logistic regression command
868 * pweight: Incorporates sampling weights at higher levels (i.e., participant level)
869 * or: Reports fixed-effects coefficients as odds ratios
870 * ##: Specifies the main effects for each variable and an interaction
```

```
* i.: Denotes a factor variable
* cmdok: Forces the "meologit" command to run on imputed data
* mi estimate: Runs the analytical model (i.e., multilevel ordinal logistic regression) within each of the imputed datasets
* 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
mi estimate, or cmdok: meologit activity2 i.education_cons##i.TimePA || idauniq:, pweight(cov19lwgtw2)
* 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
mi estimate, or cmdok: meologit activity2 i.mynssec3_cons##i.TimePA || idauniq:, pweight(cov19lwgtw2)
* 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
mi estimate, or cmdok: meologit activity2 i.wealth_cons##i.TimePA || idauniq:, pweight(cov19lwgtw2)
* 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
mi estimate, or cmdok: meologit activity2 i.education_cons##i.TimePA i.mynssec3_cons##i.TimePA i.wealth_cons##i.TimePA || idauniq:, pweight(cov19lwgtw2)
* Model 5: Two-level ordered logit regression of physical activity on indicators for levels of education, biological sex, and time, including two-way (between education and time, and between education and biological sex) and three-way (between education, biological sex, and time) interactions, with random intercepts by participant ID
mi estimate, or cmdok: meologit activity2 i.education_cons##i.TimePA i.education_cons##i.sex_cons i.education_cons#i.TimePA#i.sex_cons || idauniq:, pweight(cov19lwgtw2)
* Model 6: Two-level ordered logit regression of physical activity on indicators for levels of occupational class, biological sex, and time, including two-way (between occupational class and time, and between occupational class and biological sex) and three-way (between occupational class, biological sex, and time) interactions, with random intercepts by participant ID
mi estimate, or cmdok: meologit activity2 i.mynssec3_cons##i.TimePA i.mynssec3_cons##i.sex_cons i.mynssec3_cons#i.TimePA#i.sex_cons || idauniq:, pweight(cov19lwgtw2)
* Model 7: Two-level ordered logit regression of physical activity on indicators for levels of wealth, biological sex, and time, including two-way (between wealth and time, and between wealth and biological sex) and three-way (between wealth, biological sex, and time) interactions, with random intercepts by participant ID
mi estimate, or cmdok: meologit activity2 i.wealth_cons##i.TimePA i.wealth_cons##i.sex_cons i.wealth_cons#i.TimePA#i.sex_cons || idauniq:, pweight(cov19lwgtw2)

* FULLY ADJUSTED MODELS (I.E., WITH COVARIATES) - MULTIPLE IMPUTATION
* 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
mi estimate, or cmdok: meologit activity2 i.education_cons##i.TimePA i.sex_cons i.marital_cons i.ethnicity_cons i.employment_cons i.living_cons i.age_cons i.limiting_cons || idauniq:, pweight(cov19lwgtw2)
* 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
mi estimate, or cmdok: meologit activity2 i.mynssec3_cons##i.TimePA i.sex_cons i.marital_cons i.ethnicity_cons i.employment_cons i.living_cons i.age_cons i.limiting_cons || idauniq:, pweight(cov19lwgtw2)
* 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
mi estimate, or cmdok: meologit activity2 i.wealth_cons##i.TimePA i.sex_cons i.marital_cons i.ethnicity_cons i.employment_cons i.living_cons i.age_cons i.limiting_cons || idauniq:, pweight(cov19lwgtw2)
* 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
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.marital_cons i.ethnicity_cons i.employment_cons i.living_cons i.age_cons i.limiting_cons || idauniq:, pweight(cov19lwgtw2)
* Model 5: Two-level ordered logit regression of physical activity on indicators for levels of
```

```
      education, biological sex, and time, including two-way (between education and time, and between
      education and biological sex) and three-way (between education, biological sex, and time)
      interactions (adjusted for covariates), with random intercepts by participant ID
899   mi estimate, or cmdok: meologit activity2 i.education_cons##i.TimePA i.education_cons##i.sex_cons i.
      education_cons#i.TimePA#i.sex_cons i.marital_cons i.ethnicity_cons i.employment_cons i.living_cons i.
      age_cons i.limiting_cons || idauniq:, pweight(cov19lwgtw2)
900   * Model 6: Two-level ordered logit regression of physical activity on indicators for levels of
      occupational class, biological sex, and time, including two-way (between occupational class and
      time, and between occupational class and biological sex) and three-way (between occupational class,
      biological sex, and time) interactions (adjusted for covariates), with random intercepts by
      participant ID
901   mi estimate, or cmdok: meologit activity2 i.mynssec3_cons##i.TimePA i.mynssec3_cons##i.sex_cons i.
      mynssec3_cons#i.TimePA#i.sex_cons i.marital_cons i.ethnicity_cons i.employment_cons i.living_cons i.
      age_cons i.limiting_cons || idauniq:, pweight(cov19lwgtw2)
902   * Model 7: Two-level ordered logit regression of physical activity on indicators for levels of
      wealth, biological sex, and time, including two-way (between wealth and time, and between wealth and
      biological sex) and three-way (between wealth, biological sex, and time) interactions (adjusted for
      covariates), with random intercepts by participant ID
903   mi estimate, or cmdok: meologit activity2 i.wealth_cons##i.TimePA i.wealth_cons##i.sex_cons i.
      wealth_cons#i.TimePA#i.sex_cons i.marital_cons i.ethnicity_cons i.employment_cons i.living_cons i.
      age_cons i.limiting_cons || idauniq:, pweight(cov19lwgtw2)
```