```
**********************************************************************************************************************************************************************************************
***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 wave9par.dta

* Variables COVID Wave 2
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 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
rename CvNumP hhtotw2
rename CvMhCed_CvMhCed1_q pscedaw2
rename CvMhCed_CvMhCed2_q pscedbw2
rename CvMhCed_CvMhCed3_q pscedcw2
rename CvMhCed_CvMhCed4_q psceddw2
rename CvMhCed_CvMhCed5_q pscedew2
rename CvMhCed_CvMhCed6_q pscedfw2
rename CvMhCed_CvMhCed7_q pscedgw2
rename CvMhCed_CvMhCed8_q pscedhw2
rename HEILL heillw2
rename HELIM helimw2
rename Sex Sexw2
* 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 covidwave2par.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 wave9derivedr.dta

* Variables Wave 9 Financial Derived
use idauniq nettotw_bu_s nettotw_bu_f nettotw_bu_t 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 wave9financialr.dta

* Wave 9 complete data
* Merge core, derived, and financial datasets for Wave 9 using the participant ID
* Use Wave 9 core dataset
use wave9par.dta
* One-to-one merge of data in memory with wave9financialr.dta on participant ID
merge 1:1 idauniq using wave9financialr.dta, generate (merge_financial9)
* Overwrite Wave 9 dataset, by replacing the previously saved file
save wave9par.dta, replace
* Use the newly saved file for Wave 9
use wave9par.dta
* One-to-one merge of data in memory with wave9derivedr.dta on participant ID
merge 1:1 idauniq using wave9derivedr.dta, generate (merge_derived9)
* Sort from lowest to highest participant ID
sort idauniq
* Overwrite Wave 9 dataset, by replacing the previously saved file
save wave9par.dta, replace

* Append Wave 9 and COVID Wave 2 datasets
use wave9par.dta
append using covidwave2par.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
unique idauniq if wave==9
* 8,736 individuals
unique idauniq if wave==11
* 6,794 individuals
* Check types of sample members
tab samptyp
tab FinStat
* 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)
```

```stata
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

* Generate a new variable duplicating the age variable at Wave 9
gen indager_cons = indager 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 indager_cons, replace
* Drop observation if the participant is aged less than 60 years at Wave 9
drop if indager_cons < 60
* Count total number of participants and observations
unique idauniq
* 4,407 individuals, 8,814 observations
* Save dataset with a new name
save datapar.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
* BU total net (non-pension) wealth (Wave 9)
* Replace variable as missing for any missing cases (coded as negative numbers in the ELSA dataset)
replace nettotw_bu_s = . if inlist(nettotw_bu_s,-999,-998,-995)
* Redefine quintiles for our sample of interest
xtile quintile = nettotw_bu_s if nettotw_bu_s!=., n(5)
* 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

* Ethnicity (Wave 9)
* Replace variable as missing for any missing cases (coded as negative numbers in the ELSA dataset)
replace fqethnmr = . if fqethnmr<0
* Assign the number 0 if the participant is White
replace fqethnmr = 0 if fqethnmr == 1
* Assign the number 1 if the participant is Non-White
replace fqethnmr = 1 if fqethnmr == 2
* Coding of the final ethnicity variable:
* 0: White, 1: Non-White

* Number of people in household (Wave 9)
* Replace variable as missing for any missing cases (coded as negative numbers or 0 in the ELSA dataset)
replace hhtot = . if hhtot<0
replace hhtot = . if hhtot==0
* Assign the number 0 if one person lives in household
replace hhtot = 0 if hhtot==1
* Assign the number 1 if more than one person lives in household
replace hhtot = 1 if hhtot>1 & hhtot != .
* Coding of the final living status variable:
* 0: Living alone, 1: Not living alone

* Age categorical (Wave 9)
* Generate a new variable
gen age_cat = .
* Assign the number 0 for participants aged 60-69 years at Wave 9
replace age_cat = 0 if indager >= 60 & indager <= 69
* Assign the number 1 for participants aged 70-79 years at Wave 9
replace age_cat = 1 if indager >= 70 & indager <= 79
* Assign the number 2 for participants aged 80+ years at Wave 9 and without missing age data
replace age_cat = 2 if indager >= 80 & indager != .
* Coding of the final categorical age variable:
* 0: 60-69 years
* 1: 70-79 years
* 2: 80+ years

* Limiting long-standing illness (Wave 9)
* 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
gen limiting = 0 if heill == 2 | helim == 2
* Assign the number 1 for participants with a limiting long-standing illness
replace limiting = 1 if helim == 1
* Coding of the final limiting long-standing illness variable:
* 0: No long-standing illness or not limiting, 1: Limiting long-standing illness

* Save dataset with a new name
save data01par.dta

* 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
* 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 = quintile 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 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 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

* Save dataset with a new name
save data02par.dta

* Shielding (COVID Wave 2)
* Replace variables as missing for any missing cases (coded as negative numbers in the ELSA dataset)
replace CvStayD1 = . if CvStayD1<0
replace CvStayD2 = . if CvStayD2<0
replace CvStayD3 = . if CvStayD3<0
replace CvStayD4 = . if CvStayD4<0
replace CvStayD5 = . if CvStayD5<0
* Generate a new variable and assign the number 0 for participants who were neither self-isolating nor trying to stay at home in April
gen shielding = 0 if CvStayD3==1
* Assign the number 0 for participants who were trying to stay at home in April
replace shielding = 0 if CvStayD2==1
* Assign the number 1 for participants who were self-isolating in April
replace shielding = 1 if CvStayD1==1
* Assign the shielding value to all observations for a participant
bysort idauniq(wave): replace shielding = shielding[2]
* Coding of the final shielding variable:
* 0: Not shielding
* 1: Shielding at all times

* 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 data03par.dta

* Keep variables required for analyses
keep idauniq wave TimePA indager edqual education_cons mynssec3_cons wealth_cons sex_cons ethnicity_cons living_cons age_cons limiting_cons activity2 cov19lwgtw2 health_cons smoking_cons alcohol_cons depression_cons shielding
* 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 ethnicity_cons living_cons age_cons limiting_cons TimePA health_cons smoking_cons alcohol_cons depression_cons shielding

* Display the correlation matrix for the variables in the dataset
pwcorr, star(.05)
* Save dataset with a new name
save CCpadatar.dta

* 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, ethnicity, living status, or categorical age data are missing
drop if sex_cons == . | ethnicity_cons == . | living_cons == . | age_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 limiting long-standing illness, depressive symptoms, or shielding data are missing
drop if limiting_cons == . | depression_cons == . | shielding == .
* 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
* 3,720 individuals, 7,440 observations

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

* Create dummy variables for the physical activity outcome for hierarchical logistic regression analyses
* Inactive (coded as 0) versus mild, moderate, or vigorous (coded as 1)
gen acti0 = 1 if inlist(activity2,1,2,3)
replace acti0 = 0 if activity2 == 0
* Inactive, mild, or moderate (coded as 0) versus vigorous (coded as 1)
gen acti3 = 0 if inlist(activity2,0,1,2)
replace acti3 = 1 if activity2 == 3
* Save dataset with a new name
save ccpadummyr.dta

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

* HIERARCHICAL LOGISTIC REGRESSION
* Display base levels of factor variables and their interactions in output tables
set showbaselevels on

* UNADJUSTED MODELS
* melogit: Multilevel mixed-effects 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 logistic regression of physical activity on indicators for levels of education and time, and their interaction, with random intercepts by participant ID
melogit acti0 i.education_cons##i.TimePA || idauniq:, pweight(cov19lwgtw2) or
melogit acti3 i.education_cons##i.TimePA || idauniq:, pweight(cov19lwgtw2) or

```

```
* Model 2: Two-level logistic regression of physical activity on indicators for levels of occupational class and time, and their interaction, with random intercepts by participant ID
melogit acti0 i.mynssec3_cons##i.TimePA || idauniq:, pweight(cov19lwgtw2) or
melogit acti3 i.mynssec3_cons##i.TimePA || idauniq:, pweight(cov19lwgtw2) or

* Model 3: Two-level logistic regression of physical activity on indicators for levels of wealth and time, and their interaction, with random intercepts by participant ID
melogit acti0 i.wealth_cons##i.TimePA || idauniq:, pweight(cov19lwgtw2) or
melogit acti3 i.wealth_cons##i.TimePA || idauniq:, pweight(cov19lwgtw2) or

* Model 4: Two-level logistic 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
melogit acti0 i.education_cons##i.TimePA i.mynssec3_cons##i.TimePA i.wealth_cons##i.TimePA || idauniq:, pweight(cov19lwgtw2) or
melogit acti3 i.education_cons##i.TimePA i.mynssec3_cons##i.TimePA i.wealth_cons##i.TimePA || idauniq:, pweight(cov19lwgtw2) or

* FULLY ADJUSTED MODELS (I.E., WITH COVARIATES)
* Model 1: Two-level logistic regression of physical activity on indicators for levels of education and time, and their interaction (adjusted for covariates), with random intercepts by participant ID
melogit acti0 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
melogit acti3 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

* Model 2: Two-level logistic 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
melogit acti0 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
melogit acti3 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

* Model 3: Two-level logistic regression of physical activity on indicators for levels of wealth and time, and their interaction (adjusted for covariates), with random intercepts by participant ID
melogit acti0 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
melogit acti3 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

* Model 4: Two-level logistic 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
melogit acti0 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
melogit acti3 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
clear

* GENERALISED LINEAR MIXED MODELS
* Use complete case dataset
use CCpadatafull.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

* FULLY ADJUSTED MODELS (I.E., WITH COVARIATES)
* 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
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
* 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
margins i.education_cons i.TimePA i.education_cons#i.TimePA, predict (mu fixedonly outcome(0)) vsquish vce(unconditional)
* 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
margins i.education_cons i.TimePA i.education_cons#i.TimePA, predict (mu fixedonly outcome(1)) vsquish vce(unconditional)
* 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
margins i.education_cons i.TimePA i.education_cons#i.TimePA, predict (mu fixedonly outcome(2)) vsquish vce(unconditional)
* 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
margins i.education_cons i.TimePA i.education_cons#i.TimePA, predict (mu fixedonly outcome(3)) vsquish vce(unconditional)
* 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
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)

* 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
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
* 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 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
margins i.mynssec3_cons i.TimePA i.mynssec3_cons#i.TimePA, predict (mu fixedonly outcome(0)) vsquish vce(unconditional)
* 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.ethnicity_cons i.living_cons i.age_cons i.limiting_cons depression_cons i.shielding || 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.ethnicity_cons i.living_cons i.age_cons i.limiting_cons depression_cons i.shielding || idauniq:, pweight(cov19lwgtw2) or

* GENERALISED LINEAR LATENT AND MIXED MODELS

* Dummy variables
* Education
* Medium education (i.e., school qualifications) (coded as 1) versus low (i.e., no formal qualifications) or high (i.e., higher education) education (coded as 0)
gen mediumed = 0 if inlist(education_cons,0,2)
replace mediumed = 1 if education_cons == 1
* High education (coded as 1) versus low or medium education (coded as 0)
gen highed = 0 if inlist(education_cons,0,1)
replace highed = 1 if education_cons == 2
* Occupational class
* Intermediate occupations (coded as 1) versus lower or higher occupations (coded as 0)
gen mediumocc = 0 if inlist(mynssec3_cons,0,2)
replace mediumocc = 1 if mynssec3_cons == 1
* Higher occupations (coded as 1) versus lower or intermediate occupations (coded as 0)
gen highocc = 0 if inlist(mynssec3_cons,0,1)
replace highocc = 1 if mynssec3_cons == 2
* Wealth
* 2nd quintile (coded as 1) versus 1st, 3rd, 4th, or 5th quintile (coded as 0)
gen quint2 = 0 if inlist(wealth_cons,1,3,4,5)
replace quint2 = 1 if wealth_cons == 2
* 3rd quintile (coded as 1) versus 1st, 2nd, 4th, or 5th quintile (coded as 0)
gen quint3 = 0 if inlist(wealth_cons,1,2,4,5)
replace quint3 = 1 if wealth_cons == 3
* 4th quintile (coded as 1) versus 1st, 2nd, 3rd, or 5th quintile (coded as 0)
gen quint4 = 0 if inlist(wealth_cons,1,2,3,5)
replace quint4 = 1 if wealth_cons == 4
* 5th quintile (coded as 1) versus 1st, 2nd, 3rd, or 4th quintile (coded as 0)
gen quint5 = 0 if inlist(wealth_cons,1,2,3,4)
replace quint5 = 1 if wealth_cons == 5

* Generate interaction terms
gen int_meded = mediumed*TimePA
gen int_hied = highed*TimePA
gen int_medocc = mediumocc*TimePA
gen int_hiocc = highocc*TimePA
gen int_qui2 = quint2*TimePA
gen int_qui3 = quint3*TimePA
gen int_qui4 = quint4*TimePA
gen int_qui5 = quint5*TimePA

* Save dataset with a new name
save gllammr.dta

* UNADJUSTED MODELS
* gllamm: Generalised linear latent and mixed model command
* i(): Gives the variable that identifies the clusters
* link(ologit): ordinal logit link
* adapt: Specifies adaptive quadrature
* eform: Reports fixed-effects coefficients as odds ratios
* Model 1
* Estimation of proportional odds model
gllamm activity2 mediumed highed TimePA int_meded int_hied, i(idauniq) link(ologit) pweight(cov19lwgtw) adapt
gllamm, eform
estimates store model1
* Relax proportional odds assumption for education
eq thr: mediumed highed
estimates restore model1
matrix a=e(b)
gllamm activity2 TimePA int_meded int_hied, i(idauniq) link(ologit) thresh(thr) from(a) pweight(cov19lwgtw) skip adapt
estimates store model2
* Likelihood ratio test
lrtest model1 model2
```

```

* Model 2
* Estimation of proportional odds model
gllamm activity2 mediumocc highocc TimePA int_medocc int_hiocc, i(idauniq) link(ologit) pweight(cov19lwgtw) adapt
gllamm, eform
estimates store model3
* Relax proportional odds assumption for occupational class
eq thr: mediumocc highocc
estimates restore model3
matrix a=e(b)
gllamm activity2 TimePA int_medocc int_hiocc, i(idauniq) link(ologit) thresh(thr) from(a) pweight(cov19lwgtw) skip adapt
estimates store model4
* Likelihood ratio test
lrtest model3 model4

* Model 3
* Estimation of proportional odds model
gllamm activity2 quint2 quint3 quint4 quint5 TimePA int_qui2 int_qui3 int_qui4 int_qui5, i(idauniq) link(ologit) pweight(cov19lwgtw) adapt
gllamm, eform
estimates store model5
* Relax proportional odds assumption for wealth
eq thr: quint2 quint3 quint4 quint5
estimates restore model5
matrix a=e(b)
gllamm activity2 TimePA int_qui2 int_qui3 int_qui4 int_qui5, i(idauniq) link(ologit) thresh(thr) from(a) pweight(cov19lwgtw) skip adapt
estimates store model6
* Likelihood ratio test
lrtest model5 model6
clear

* Descriptive statistics
* Use participant ID and cross-sectional weight from Wave 9 core dataset
use idauniq w9xwgt 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 wave9crossweightr.dta

* Use participant ID and cross-sectional weight from COVID Wave 2 dataset
use idauniq wtfin1 using elsa_covid_w2_eul.dta
* Describe dataset
describe
* Sort from lowest to highest participant ID
sort idauniq
* Generate a new variable called wave and assign the number 11 to each observation (to designate COVID Wave 2)
gen wave=11
* Save dataset with a new name
save wave11crossweightr.dta

* Use complete case dataset
use CCpadatafullr.dta
* One-to-one merge of data in memory with wave9crossweightr.dta on participant ID and wave
merge 1:1 idauniq wave using wave9crossweightr.dta, generate(merge_crossweight9)
```

```
* Sort from lowest to highest participant ID
sort idauniq
* Drop observations for which the key variable (participant ID) does not match
drop if merge_crossweight9==2
* Sort from lowest to highest participant ID and wave
sort idauniq wave
* One-to-one merge of data in memory with wave11crossweightr.dta on participant ID and wave
merge 1:1 idauniq wave using wave11crossweightr.dta, generate(merge_crossweight11)
* Sort from lowest to highest participant ID
sort idauniq
* Drop observations for which the key variable (participant ID) does not match
drop if merge_crossweight11==2
* Sort from lowest to highest participant ID and wave
sort idauniq wave

* 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
* Tables of frequencies - COVID Wave 2
tab activity2 if wave==11
tab activity2 if wave==11 [aw=wtfin1]
tab shielding if wave==11
tab shielding if wave==11 [aw=wtfin1]
tab activity2 if wave==11 & wtfin1==.
tab shielding if wave==11 & wtfin1==.
* Keep data from Wave 9 (baseline) only
keep if wave==9
* Save dataset with a new name
save descw9padatar.dta

* Tables of frequencies, weighted using the Wave 9 cross-sectional weight
tab activity2 [aw=w9xwgt]
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]
tab living_cons [aw=w9xwgt]
tab limiting_cons [aw=w9xwgt]
* Summary statistics for continuous variables, weighted using the Wave 9 cross-sectional weight
sum indager [aw=w9xwgt]
sum depression_cons [aw=w9xwgt]

* Tables of frequencies, unweighted
tab activity2
tab education_cons
tab mynssec3_cons
tab wealth_cons
tab sex_cons
tab ethnicity_cons
tab age_cons
tab living_cons
tab limiting_cons
* Summary statistics for continuous variables, unweighted
sum indager
sum depression_cons

* Use dataset with processed variables
use data03par.dta
* Keep variables required for analyses and multiple imputation
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
* Count total number of participants and observations
unique idauniq
* 4,407 individuals, 8,814 observations

* Time-constant education (raw) - Wave 9
* Generate a new variable duplicating the raw education variable at Wave 9
gen edqual_cons = edqual 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 edqual_cons, replace

* Time-constant occupational class (raw) - Wave 9
* Generate a new variable duplicating the raw occupational class variable at Wave 9
gen nssec8_cons = nssec8 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 nssec8_cons, replace

* Assign extended missing value for response "foreign/other" for education
replace education_cons = .a if edqual_cons==6
* Assign extended missing value for response "never worked and long-term unemployed" for occupational class
replace mynssec3_cons = .a if nssec8_cons==8
* Drop unnecessary variables
drop nssec8 edqual nssec8_cons edqual_cons mynssec3 education

* 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 ethnicity_cons living_cons age_cons limiting_cons TimePA health_cons depression_cons alcohol_cons smoking_cons shielding
* Save dataset with a new name
save imputationr.dta

* 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 ethnicity_cons living_cons age_cons limiting_cons TimePA health_cons depression_cons alcohol_cons smoking_cons shielding
* Display patterns of missing data
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
* 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 ethnicity_cons living_cons age_cons limiting_cons TimePA health_cons depression_cons alcohol_cons smoking_cons shielding, generate(miss_)
* Review changes
describe miss_*

* 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
ologit activity2 i.health_cons i.alcohol_cons i.smoking_cons
logit miss_activity2 i.health_cons i.alcohol_cons i.smoking_cons
ologit education_cons i.health_cons i.alcohol_cons i.smoking_cons
```

```
logit miss_education_cons i.health_cons i.alcohol_cons i.smoking_cons if education_cons <= .
ologit mynssec3_cons i.health_cons i.alcohol_cons i.smoking_cons
logit miss_mynssec3_cons i.health_cons i.alcohol_cons i.smoking_cons if mynssec3_cons <= .
ologit wealth_cons i.health_cons i.alcohol_cons i.smoking_cons
logit miss_wealth_cons i.health_cons i.alcohol_cons i.smoking_cons
logit sex_cons i.health_cons i.alcohol_cons i.smoking_cons
logit ethnicity_cons i.health_cons i.alcohol_cons i.smoking_cons
logit living_cons i.health_cons i.alcohol_cons i.smoking_cons
ologit age_cons i.health_cons i.alcohol_cons i.smoking_cons
logit limiting_cons i.health_cons i.alcohol_cons i.smoking_cons
logit miss_limiting_cons i.health_cons i.alcohol_cons i.smoking_cons
regress depression_cons i.health_cons i.alcohol_cons i.smoking_cons
logit miss_depression_cons i.health_cons i.alcohol_cons i.smoking_cons
logit shielding i.health_cons i.alcohol_cons i.smoking_cons
logit miss_shielding i.health_cons i.alcohol_cons i.smoking_cons

* Drop unnecessary variables
drop miss_* wave
* Reshape data into wide format for observations identified by participant ID and add "TimePA" as an identifying time period
mi reshape wide activity2, i(idauniq) j(TimePA)
* Register all variables with missing values that need to be imputed
mi register imputed activity20 activity21 education_cons mynssec3_cons wealth_cons limiting_cons health_cons depression_cons alcohol_cons shielding
* Register all variables with no missing values and/or which do not require imputation
mi register regular sex_cons ethnicity_cons living_cons age_cons smoking_cons
* Clear panel data settings
mi xtset, clear

* Impute variables
* Imputation methods:
* ologit: ordinal logistic
* logit: logistic
* nbreg: negative binomial regression
* 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.
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
* Save the multiple datasets in wide format
save imputedwidepaar.dta

* Reshape data into long format
mi reshape long activity2, i(idauniq) j(TimePA)
* Save the multiple datasets in long format
save imputedlongpaar.dta

* GENERALISED LINEAR MIXED MODELS - MULTIPLE IMPUTATION
* Use multiply imputed dataset in long format
use imputedlongpaar.dta
* Display base levels of factor variables and their interactions in output tables
set showbaselevels on

* UNADJUSTED MODELS - MULTIPLE IMPUTATION
* 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
* 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)

* 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.ethnicity_cons i.living_cons i.age_cons i.limiting_cons depression_cons i.shielding || 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.ethnicity_cons i.living_cons i.age_cons i.limiting_cons depression_cons i.shielding || 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.ethnicity_cons i.living_cons i.age_cons i.limiting_cons depression_cons i.shielding || 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.ethnicity_cons i.living_cons i.age_cons i.limiting_cons depression_cons i.shielding || idauniq:, pweight(cov19lwgtw2)
```