WEB MATERIAL

Web Appendix 1

ANNOTATED STATA (VERSION 14.1) CODE FOR ALL ANALYSES

```
set more off

* Read in merged public-use Wave 1 and Wave 2 NESARC data, including computed variables and Wave
* 2 nonresponse indicator (nonresp).
* (SAS code used to generate the computed variables and perform the merging is available upon
* request.)

use "C:\w1w2_oct14r.dta", clear

* Set the sample design features as being those from Wave 1 (initially).
svyset psu [pweight = weight], strata(stratum) singleunit(centered)

* Create numeric version of race / ethnicity.
destring ETHRACE2A, generate(ethrace2a)

* For handling missing data on marijuana use.
replace S3BQ1A5 = 2 if S3BQ1A5 == .
replace S3BD5Q2B = 2 if S3BD5Q2B == .
replace W2S3BQ1A5 = 2 if W2S3BQ1A5 == . & nonresp == 0
replace W2S3BD5Q2A = 2 if W2S3BD5Q2A == . & nonresp == 0
replace S3BQ1A5 = . if S3BQ1A5 == 9
replace S3BD5Q2B = . if S3BD5Q2B == 9
replace W2S3BQ1A5 = . if W2S3BQ1A5 == 9
replace W2S3BD5Q2A = . if W2S3BD5Q2A == 9

* Compute indicator for young adult subpopulation.
gen young = 1 if age >= 18 & age <= 24
replace young = 0 if young == .

* Descriptive statistics for young indicator.
tab young
tab young nonresp, row

svy, subpop(if young == 1): tab ETHRACE2A
svy, subpop(if young == 1): tab sex

* Generate estimates reported in Table 1.

tab smoker nonresp if young == 1, row
svy, subpop(young): tab smoker nonresp, row obs

tab consumer nonresp if young == 1, row
svy, subpop(young): tab consumer nonresp, row obs

tab mjuse_life nonresp if young == 1, row
svy, subpop(young): tab mjuse_life nonresp, row obs

gen anyill_plusmj = 1 if dgstatus == 3
replace anyill_plusmj = 2 if ckstatus == 2
replace anyill_plusmj = 3 if ckstatus == 1 | ckstatus == 3

tab anyill_plusmj

tab anyill_plusmj nonresp if young == 1, row
svy, subpop(young): tab anyill_plusmj nonresp, row obs

* Generate estimates reported in Table 2.

tab nic_life nonresp if young == 1, row
svy, subpop(young): tab nic_life nonresp, row obs

tab life_aud nonresp if young == 1, row
```

```
svy, subpop(young): tab life_aud nonresp, row obs

tab life_mjud nonresp if young == 1, row
svy, subpop(young): tab life_mjud nonresp, row obs

* Generate estimates reported in Table 3.

tab life_alctx nonresp if young == 1, row
svy, subpop(young): tab life_alctx nonresp, row obs

tab life_alcmt nonresp if young == 1, row
svy, subpop(young): tab life_alcmt nonresp, row obs

tab life_drugtx nonresp if young == 1, row
svy, subpop(young): tab life_drugtx nonresp, row obs

tab life_drugmt nonresp if young == 1, row
svy, subpop(young): tab life_drugmt nonresp, row obs

* Generate weighted complete-case estimates in Table 4.

gen py_tob_chg = 1 if (smoker == 2 | smoker == 3) & (ltsmoker == 2 | ltsmoker == 3 | ltsmoker ==
4)
replace py_tob_chg = 2 if smoker == 1 & (ltsmoker == 2 | ltsmoker == 3 | ltsmoker == 4)
replace py_tob_chg = 3 if (smoker == 2 | smoker == 3) & ltsmoker == 1
replace py_tob_chg = 4 if smoker == 1 & ltsmoker == 1
tab py_tob_chg if young == 1, miss

svy, subpop(young): prop py_tob_chg

gen py_alc_chg = 1 if (consumer == 2 | consumer == 3) & (W2CONSUMER == 2 | W2CONSUMER == 3)
replace py_alc_chg = 2 if consumer == 1 & (W2CONSUMER == 2 | W2CONSUMER == 3)
replace py_alc_chg = 3 if (consumer == 2 | consumer == 3) & W2CONSUMER == 1
replace py_alc_chg = 4 if consumer == 1 & W2CONSUMER == 1
tab py_alc_chg if young == 1, miss

svy, subpop(young): prop py_alc_chg

gen w1mj = 0 if S3BQ1A5 == 2 | S3BD5Q2B == 2
replace w1mj = 1 if S3BD5Q2B == 1 | S3BD5Q2B == 3
tab w1mj, miss
gen w2mj = 0 if W2S3BQ1A5 == 2 | W2S3BD5Q2A == 2
replace w2mj = 1 if W2S3BD5Q2A == 1
tab w2mj, miss

gen py_mj_chg2 = 1 if w1mj == 0 & w2mj == 0
replace py_mj_chg2 = 2 if w1mj == 1 & w2mj == 0
replace py_mj_chg2 = 3 if w1mj == 0 & w2mj == 1
replace py_mj_chg2 = 4 if w1mj == 1 & w2mj == 1
replace py_mj_chg2 = 1 if ((dgstatus == 2 | dgstatus == 3) & (W2DGSTATUS == 2 | W2DGSTATUS == 3))
| ltdgstatus == 4
tab py_mj_chg2 if young == 1, miss

svy, subpop(young): prop py_mj_chg2

gen py_nicd_chg = 1 if TAB12MDX == 0 & curtabdep == 0
replace py_nicd_chg = 2 if TAB12MDX == 1 & curtabdep == 0
replace py_nicd_chg = 3 if TAB12MDX == 0 & curtabdep == 1
replace py_nicd_chg = 4 if TAB12MDX == 1 & curtabdep == 1
tab py_nicd_chg if young == 1, miss

svy, subpop(young): prop py_nicd_chg

gen py_alcd_chg = 1 if ALCABDEP12DX == 0 & W2AAD12 == 0
replace py_alcd_chg = 2 if (ALCABDEP12DX >= 1 & ALCABDEP12DX <= 3) & (W2AAD12 == 0)
replace py_alcd_chg = 3 if ALCABDEP12DX == 0 & (W2AAD12 >= 1 & W2AAD12 <= 3)
replace py_alcd_chg = 4 if (ALCABDEP12DX >= 1 & ALCABDEP12DX <= 3) & (W2AAD12 >= 1 & W2AAD12 <=
3)
tab py_alcd_chg if young == 1, miss

svy, subpop(young): prop py_alcd_chg
```

```
gen py_mjud_chg = 1 if MAR12ABDEP == 0 & MAABDEP12 == 0
replace py_mjud_chg = 2 if MAR12ABDEP >= 1 & MAR12ABDEP <= 3 & MAABDEP12 == 0
replace py_mjud_chg = 3 if MAR12ABDEP == 0 & MAABDEP12 >= 1 & MAABDEP12 <= 3
replace py_mjud_chg = 4 if MAR12ABDEP >= 1 & MAR12ABDEP <= 3 & MAABDEP12 >= 1 & MAABDEP12 <= 3
tab py_mjud_chg if young == 1, miss

svy, subpop(young): prop py_mjud_chg

* Alternative way of coding marijuana use disorder change (same results).

gen mjud1 = 0 if MAR12ABDEP == 0
replace mjud1 = 1 if MAR12ABDEP >= 1 & MAR12ABDEP <= 3
gen mjud2 = 0 if MAABDEP12 == 0
replace mjud2 = 1 if MAABDEP12 <= 3 & MAABDEP12 >= 1

gen py_mjud_chg2 = 1 if mjud1 == 0 & mjud2 == 0
replace py_mjud_chg2 = 2 if mjud1 == 1 & mjud2 == 0
replace py_mjud_chg2 = 3 if mjud1 == 0 & mjud2 == 1
replace py_mjud_chg2 = 4 if mjud1 == 1 & mjud2 == 1

svy, subpop(young): prop py_mjud_chg2

* For Table 4, repeat all analyses above after using adjusted Wave 2 weights and Wave 2 design
variables.
svyset W2PSU [pweight = W2WEIGHT], strata(W2STRATUM) singleunit(centered)


*******************************************************
* Ordered probit selection model approach for Table 4. *
*******************************************************

*********************
* py_tob_chg outcome
*********************

* Compute income tertiles for young adults.
tab S1Q10B if young == 1
gen income2 = 1 if S1Q10B >= 0 & S1Q10B <= 1
replace income2 = 2 if S1Q10B >= 2 & S1Q10B <= 5
replace income2 = 3 if S1Q10B >= 6 & S1Q10B != .
tab income2 if young == 1, miss

* Compute response indicator (as opposed to nonresponse indicator)
gen resp = 1 if nonresp == 0
replace resp = 0 if nonresp == 1

* Note: substantive model is consistent with predictors in imputation models.
* Note: selection model is based on McCabe and West (in press).
* Marijuana use at Wave 1 was not considered as a covariate in the selection model
* because it was not fully observed (a limitation of the selection model approach).

svy: heckoprobit py_tob_chg age i.sex i.ethrace2a i.educate i.income2 i.smoker i.consumer ///
     any_py_anxd any_py_moodd any_lt_persd, ///
        select(resp = i.consumer age i.sex i.ethrace2a i.educate i.income2)

* Best explanatory variables (sig. in both equations): age, sex, race, education, income
* Instrument: consumer (not sig. in substantive equation)

* Note: Minimal evidence of selection bias for py_tob_chg:
* rho |  -.0468146    .019159                      -.0849905   -.0085014

* Calculate predicted probabilities of each outcome (py_tob_chg) category,
* conditional on being a non-respondent.
predict p1, outcome(1) pcond0
predict p2, outcome(2) pcond0
predict p3, outcome(3) pcond0
predict p4, outcome(4) pcond0

* Implement Clark and Houle (2012) method to compute adjusted estimates.
gen cat1 = 1 if py_tob_chg == 1 & resp == 1
```

```
replace cat1 = 0 if py_tob_chg != 1 & py_tob_chg != . & resp == 1
gen cat2 = 1 if py_tob_chg == 2 & resp == 1
replace cat2 = 0 if py_tob_chg != 2 & py_tob_chg != . & resp == 1
gen cat3 = 1 if py_tob_chg == 3 & resp == 1
replace cat3 = 0 if py_tob_chg != 3 & py_tob_chg != . & resp == 1
gen cat4 = 1 if py_tob_chg == 4 & resp == 1
replace cat4 = 0 if py_tob_chg != 4 & py_tob_chg != . & resp == 1
replace cat1 = p1 if resp == 0
replace cat2 = p2 if resp == 0
replace cat3 = p3 if resp == 0
replace cat4 = p4 if resp == 0
svyset psu [pweight = weight], strata(stratum) singleunit(centered)
svy, subpop(young): mean cat1
svy, subpop(young): mean cat2
svy, subpop(young): mean cat3
svy, subpop(young): mean cat4

drop p1 p2 p3 p4 cat1 cat2 cat3 cat4

********************
* py_alc_chg outcome
********************

svy: heckoprobit py_alc_chg age i.sex i.ethrace2a i.educate i.income2 i.smoker i.consumer ///
     any_py_anxd any_py_moodd any_lt_persd, ///
        select(resp = i.consumer age i.sex i.ethrace2a i.educate i.income2 any_py_moodd )

* Best explanatory variables (sig. in both equations): age, sex, race, education, income,
consumer
* Instrument: any_py_moodd (not sig. in substantive equation)

* Note: Moderate evidence of selection bias for py_alc_chg:
* rho |   .1012515   .0167329                      .0677317     .134543

* Calculate predicted probabilities of each outcome (py_alc_chg) category,
* conditional on being a non-respondent.
predict p1, outcome(1) pcond0
predict p2, outcome(2) pcond0
predict p3, outcome(3) pcond0
predict p4, outcome(4) pcond0

* Implement Clark and Houle (2012) method to compute adjusted estimates.
gen cat1 = 1 if py_alc_chg == 1 & resp == 1
replace cat1 = 0 if py_alc_chg != 1 & py_alc_chg != . & resp == 1
gen cat2 = 1 if py_alc_chg == 2 & resp == 1
replace cat2 = 0 if py_alc_chg != 2 & py_alc_chg != . & resp == 1
gen cat3 = 1 if py_alc_chg == 3 & resp == 1
replace cat3 = 0 if py_alc_chg != 3 & py_alc_chg != . & resp == 1
gen cat4 = 1 if py_alc_chg == 4 & resp == 1
replace cat4 = 0 if py_alc_chg != 4 & py_alc_chg != . & resp == 1
replace cat1 = p1 if resp == 0
replace cat2 = p2 if resp == 0
replace cat3 = p3 if resp == 0
replace cat4 = p4 if resp == 0
svyset psu [pweight = weight], strata(stratum) singleunit(centered)
svy, subpop(young): mean cat1
svy, subpop(young): mean cat2
svy, subpop(young): mean cat3
svy, subpop(young): mean cat4

drop p1 p2 p3 p4 cat1 cat2 cat3 cat4


********************
* py_mj_chg2 outcome
********************

* NOTE: 11 missing values at Wave 1 assumed to be "No" for Marijuana Use (W1MJ), for
* purposes of maintaining constant sample sizes

gen w1mj_rec = w1mj if w1mj != .
```

```
replace w1mj_rec = 0 if w1mj == .
tab w1mj_rec

svy: heckoprobit py_mj_chg2 age i.sex i.ethrace2a i.educate i.income2 w1mj_rec /* i.consumer */
///
     any_py_anxd any_py_moodd any_lt_persd, ///
        select(resp = i.consumer age i.sex i.ethrace2a i.educate i.income2)

* Best explanatory variables (sig. in both equations): age, sex, race
* Instruments: consumer, education, income (not sig. in substantive equation)

* Note: Moderate evidence of selection bias for py_mj_chg2:
* rho |  -.4109774   .0548556                      -.5143342   -.2958525

* Calculate predicted probabilities of each outcome (py_mj_chg2) category,
* conditional on being a non-respondent.
predict p1, outcome(1) pcond0
predict p2, outcome(2) pcond0
predict p3, outcome(3) pcond0
predict p4, outcome(4) pcond0

* Implement Clark and Houle (2012) method to compute adjusted estimates.
gen cat1 = 1 if py_mj_chg2 == 1 & resp == 1
replace cat1 = 0 if py_mj_chg2 != 1 & py_mj_chg2 != . & resp == 1
gen cat2 = 1 if py_mj_chg2 == 2 & resp == 1
replace cat2 = 0 if py_mj_chg2 != 2 & py_mj_chg2 != . & resp == 1
gen cat3 = 1 if py_mj_chg2 == 3 & resp == 1
replace cat3 = 0 if py_mj_chg2 != 3 & py_mj_chg2 != . & resp == 1
gen cat4 = 1 if py_mj_chg2 == 4 & resp == 1
replace cat4 = 0 if py_mj_chg2 != 4 & py_mj_chg2 != . & resp == 1
replace cat1 = p1 if resp == 0
replace cat2 = p2 if resp == 0
replace cat3 = p3 if resp == 0
replace cat4 = p4 if resp == 0
svyset psu [pweight = weight], strata(stratum) singleunit(centered)
svy, subpop(young): mean cat1
svy, subpop(young): mean cat2
svy, subpop(young): mean cat3
svy, subpop(young): mean cat4

drop p1 p2 p3 p4 cat1 cat2 cat3 cat4


*******************************************************************************
* py_nicd_chg outcome (note that predictors have been updated to be consistent *
* with imputation model used to impute disorder outcomes)                      *
*******************************************************************************

svy: heckoprobit py_nicd_chg age i.sex i.ethrace2a i.educate i.income2 i.TAB12MDX i.ALCABDEP12DX
mjud1 ///
     any_py_anxd any_py_moodd any_lt_persd, ///
        select(resp = i.consumer age i.sex i.ethrace2a i.educate i.income2)

* Best explanatory variables (sig. in both equations): age, sex, race, education, income
* Instrument: consumer (not in substantive equation)

* Note: Moderate evidence of selection bias for py_nicd_chg:
* rho |  -.3693688   .0505938                      -.4658041   -.2642627

* Calculate predicted probabilities of each outcome (py_nicd_chg) category,
* conditional on being a non-respondent.
predict p1, outcome(1) pcond0
predict p2, outcome(2) pcond0
predict p3, outcome(3) pcond0
predict p4, outcome(4) pcond0

* Implement Clark and Houle (2012) method to compute adjusted estimates.
gen cat1 = 1 if py_nicd_chg == 1 & resp == 1
replace cat1 = 0 if py_nicd_chg != 1 & py_nicd_chg != . & resp == 1
gen cat2 = 1 if py_nicd_chg == 2 & resp == 1
replace cat2 = 0 if py_nicd_chg != 2 & py_nicd_chg != . & resp == 1
```

```
gen cat3 = 1 if py_nicd_chg == 3 & resp == 1
replace cat3 = 0 if py_nicd_chg != 3 & py_nicd_chg != . & resp == 1
gen cat4 = 1 if py_nicd_chg == 4 & resp == 1
replace cat4 = 0 if py_nicd_chg != 4 & py_nicd_chg != . & resp == 1
replace cat1 = p1 if resp == 0
replace cat2 = p2 if resp == 0
replace cat3 = p3 if resp == 0
replace cat4 = p4 if resp == 0
svyset psu [pweight = weight], strata(stratum) singleunit(centered)
svy, subpop(young): mean cat1
svy, subpop(young): mean cat2
svy, subpop(young): mean cat3
svy, subpop(young): mean cat4

drop p1 p2 p3 p4 cat1 cat2 cat3 cat4


**********************
* py_alcd_chg outcome
**********************

svy: heckoprobit py_alcd_chg age i.sex i.ethrace2a i.educate i.income2 i.TAB12MDX i.ALCABDEP12DX
mjud1 ///
     any_py_anxd any_py_moodd any_lt_persd, ///
         select(resp = i.consumer age i.sex i.ethrace2a i.educate i.income2)

* Best explanatory variables (sig. in both equations): age, sex, race
* Instrument: consumer (not in substantive equation)

* Note: Strong evidence of selection bias for py_alcd_chg:
* rho |  -.5317219   .0641062                      -.647532   -.3919047

* Calculate predicted probabilities of each outcome (py_alcd_chg) category,
* conditional on being a non-respondent.
predict p1, outcome(1) pcond0
predict p2, outcome(2) pcond0
predict p3, outcome(3) pcond0
predict p4, outcome(4) pcond0

* Implement Clark and Houle (2012) method to compute adjusted estimates.
gen cat1 = 1 if py_alcd_chg == 1 & resp == 1
replace cat1 = 0 if py_alcd_chg != 1 & py_alcd_chg != . & resp == 1
gen cat2 = 1 if py_alcd_chg == 2 & resp == 1
replace cat2 = 0 if py_alcd_chg != 2 & py_alcd_chg != . & resp == 1
gen cat3 = 1 if py_alcd_chg == 3 & resp == 1
replace cat3 = 0 if py_alcd_chg != 3 & py_alcd_chg != . & resp == 1
gen cat4 = 1 if py_alcd_chg == 4 & resp == 1
replace cat4 = 0 if py_alcd_chg != 4 & py_alcd_chg != . & resp == 1
replace cat1 = p1 if resp == 0
replace cat2 = p2 if resp == 0
replace cat3 = p3 if resp == 0
replace cat4 = p4 if resp == 0
svyset psu [pweight = weight], strata(stratum) singleunit(centered)
svy, subpop(young): mean cat1
svy, subpop(young): mean cat2
svy, subpop(young): mean cat3
svy, subpop(young): mean cat4

drop p1 p2 p3 p4 cat1 cat2 cat3 cat4


**********************
* py_mjud_chg2 outcome
**********************

svy: heckoprobit py_mjud_chg2 age i.sex i.ethrace2a i.educate i.income2 i.TAB12MDX i.ALCABDEP12DX
mjud1 ///
     any_py_anxd any_py_moodd any_lt_persd, ///
         select(resp = i.consumer age i.sex i.ethrace2a i.educate i.income2)

* Best explanatory variables (sig. in both equations): age, sex, race, income
```

```
* Instrument: consumer (not in substantive equation)

* Note: Moderate evidence of selection bias for py_mjud_chg2:
* rho |  -.2778881    .047155                      -.369158   -.1813091

* Calculate predicted probabilities of each outcome (py_mjud_chg2) category,
* conditional on being a non-respondent.
predict p1, outcome(1) pcond0
predict p2, outcome(2) pcond0
predict p3, outcome(3) pcond0
predict p4, outcome(4) pcond0

* Implement Clark and Houle (2012) method to compute adjusted estimates.
gen cat1 = 1 if py_mjud_chg2 == 1 & resp == 1
replace cat1 = 0 if py_mjud_chg2 != 1 & py_mjud_chg2 != . & resp == 1
gen cat2 = 1 if py_mjud_chg2 == 2 & resp == 1
replace cat2 = 0 if py_mjud_chg2 != 2 & py_mjud_chg2 != . & resp == 1
gen cat3 = 1 if py_mjud_chg2 == 3 & resp == 1
replace cat3 = 0 if py_mjud_chg2 != 3 & py_mjud_chg2 != . & resp == 1
gen cat4 = 1 if py_mjud_chg2 == 4 & resp == 1
replace cat4 = 0 if py_mjud_chg2 != 4 & py_mjud_chg2 != . & resp == 1
replace cat1 = p1 if resp == 0
replace cat2 = p2 if resp == 0
replace cat3 = p3 if resp == 0
replace cat4 = p4 if resp == 0
svyset psu [pweight = weight], strata(stratum) singleunit(centered)
svy, subpop(young): mean cat1
svy, subpop(young): mean cat2
svy, subpop(young): mean cat3
svy, subpop(young): mean cat4

drop p1 p2 p3 p4 cat1 cat2 cat3 cat4


*********************************
* Multiple imputation analyses *
*********************************

*************************
* Use variables first... *
*************************

mi set flong
mi register imputed ltsmoker W2CONSUMER w1mj w2mj
mi register regular age sex ethrace2a educate income2 smoker consumer any_py_anxd any_py_moodd
any_lt_persd psu stratum weight
mi describe

set seed 41279

mi impute chained (mlogit) ltsmoker W2CONSUMER (logit) w1mj w2mj ///
   = age i.sex i.ethrace2a i.educate i.income2 i.smoker i.consumer ///
     any_py_anxd any_py_moodd any_lt_persd, noisily augment add(5) burnin(5)

* Check sensitivity of FMI to increasing the number of imputations to 10
mi impute chained (mlogit) ltsmoker W2CONSUMER (logit) w1mj w2mj ///
   = age i.sex i.ethrace2a i.educate i.income2 i.smoker i.consumer ///
     any_py_anxd any_py_moodd any_lt_persd, noisily augment add(10) burnin(5)

mi svyset psu [pweight = weight], strata(stratum) singleunit(centered)

drop py_tob_chg py_alc_chg py_mj_chg2

gen py_tob_chg = 1 if (smoker == 2 | smoker == 3) & (ltsmoker == 2 | ltsmoker == 3 | ltsmoker ==
4)
replace py_tob_chg = 2 if smoker == 1 & (ltsmoker == 2 | ltsmoker == 3 | ltsmoker == 4)
replace py_tob_chg = 3 if (smoker == 2 | smoker == 3) & ltsmoker == 1
replace py_tob_chg = 4 if smoker == 1 & ltsmoker == 1

gen py_alc_chg = 1 if (consumer == 2 | consumer == 3) & (W2CONSUMER == 2 | W2CONSUMER == 3)
replace py_alc_chg = 2 if consumer == 1 & (W2CONSUMER == 2 | W2CONSUMER == 3)
```

```
replace py_alc_chg = 3 if (consumer == 2 | consumer == 3) & W2CONSUMER == 1
replace py_alc_chg = 4 if consumer == 1 & W2CONSUMER == 1

gen py_mj_chg2 = 1 if w1mj == 0 & w2mj == 0
replace py_mj_chg2 = 2 if w1mj == 1 & w2mj == 0
replace py_mj_chg2 = 3 if w1mj == 0 & w2mj == 1
replace py_mj_chg2 = 4 if w1mj == 1 & w2mj == 1
replace py_mj_chg2 = 1 if ((dgstatus == 2 | dgstatus == 3) & (W2DGSTATUS == 2 | W2DGSTATUS == 3))
| ltdgstatus == 4
tab py_mj_chg2 if young == 1, miss

mi estimate, vartable: svy, subpop(young): proportion py_tob_chg
mi estimate, vartable: svy, subpop(young): proportion py_alc_chg
mi estimate, vartable nowarning: svy, subpop(young): proportion py_mj_chg2

************************
* Disorder variables... *
************************

mi set flong
mi register imputed curtabdep W2AAD12 mjud2
mi register regular age sex ethrace2a educate income2 TAB12MDX ALCABDEP12DX mjud1 any_py_anxd
any_py_moodd any_lt_persd psu stratum weight
mi describe

set seed 121080

mi impute chained (mlogit) curtabdep W2AAD12 mjud2 ///
   = age i.sex i.ethrace2a i.educate i.income2 i.TAB12MDX i.ALCABDEP12DX mjud1 ///
     any_py_anxd any_py_moodd any_lt_persd, noisily augment add(10) burnin(5)

mi svyset psu [pweight = weight], strata(stratum) singleunit(centered)

drop py_nicd_chg py_alcd_chg py_mjud_chg2

gen py_nicd_chg = 1 if TAB12MDX == 0 & curtabdep == 0
replace py_nicd_chg = 2 if TAB12MDX == 1 & curtabdep == 0
replace py_nicd_chg = 3 if TAB12MDX == 0 & curtabdep == 1
replace py_nicd_chg = 4 if TAB12MDX == 1 & curtabdep == 1
tab py_nicd_chg if young == 1, miss

mi estimate, vartable: svy, subpop(young): prop py_nicd_chg

gen py_alcd_chg = 1 if ALCABDEP12DX == 0 & W2AAD12 == 0
replace py_alcd_chg = 2 if (ALCABDEP12DX >= 1 & ALCABDEP12DX <= 3) & (W2AAD12 == 0)
replace py_alcd_chg = 3 if ALCABDEP12DX == 0 & (W2AAD12 >= 1 & W2AAD12 <= 3)
replace py_alcd_chg = 4 if (ALCABDEP12DX >= 1 & ALCABDEP12DX <= 3) & (W2AAD12 >= 1 & W2AAD12 <=
3)
tab py_alcd_chg if young == 1, miss

mi estimate, vartable: svy, subpop(young): prop py_alcd_chg

gen py_mjud_chg2 = 1 if mjud1 == 0 & mjud2 == 0
replace py_mjud_chg2 = 2 if mjud1 == 1 & mjud2 == 0
replace py_mjud_chg2 = 3 if mjud1 == 0 & mjud2 == 1
replace py_mjud_chg2 = 4 if mjud1 == 1 & mjud2 == 1

mi estimate, vartable: svy, subpop(young): prop py_mjud_chg2
```

ANNOTATED SAS (VERSION 9.4) CODE FOR MI ANALYSES USING PMMs

```
libname desk "C:\ ";

/* Replicate approach assuming ignorable MAR mechanism, as shown in Stata code (Use Variables) */

proc mi data=desk.use_forimp seed=41279 nimpute=10 out=outfcs_discrim;
   class sex ethrace2a educate income2 smoker consumer ltsmoker W2CONSUMER w1mj w2mj;
   fcs logistic (ltsmoker W2CONSUMER / link=glogit);
   fcs logistic (w1mj w2mj);
   var ltsmoker W2CONSUMER w1mj w2mj age sex ethrace2a educate
      income2 smoker consumer any_py_anxd any_py_moodd any_lt_persd;
run;

data outfcs_discrim2;
   set outfcs_discrim;
   if (smoker = 2 or smoker = 3) and (ltsmoker = 2 or ltsmoker = 3 or ltsmoker = 4) then
py_tob_chg = 1;
   else if (smoker = 1) and (ltsmoker = 2 or ltsmoker = 3 or ltsmoker = 4) then py_tob_chg = 2;
   else if (smoker = 2 or smoker = 3) and (ltsmoker = 1) then py_tob_chg = 3;
   else if smoker = 1 and ltsmoker = 1 then py tob chg = 4;
   if (consumer = 2 or consumer = 3) and (W2CONSUMER = 2 or W2CONSUMER = 3) then py_alc_chg = 1;
   else if (consumer = 1) and (W2CONSUMER = 2 or W2CONSUMER = 3) then py_alc_chg = 2;
   else if (consumer = 2 or consumer = 3) and (W2CONSUMER = 1) then py_alc_chg = 3;
   else if consumer = 1 and W2CONSUMER = 1 then py_alc_chg = 4;
   if w1mj = 0 and w2mj = 0 then py_mj_chg2 = 1;
   else if w1mj = 1 and w2mj = 0 then py mj chg2 = 2;
   else if w1mj = 0 and w2mj = 1 then py_mj_chg2 = 3;
   else if w1mj = 1 and w2mj = 1 then py_mj_chg2 = 4;
   if (18 <= age <= 24) then young = 1;
   else young = 0;
run;

proc freq data = outfcs_discrim2;
   tables young;
run;

%macro miest(varname);
proc surveyfreq data=outfcs_discrim2;
   weight weight;
   cluster psu;
   stratum stratum;
   tables _imputation_*young*&varname / row;
   ods output CrossTabs = temptab;
run;

data temptab2 (keep = _Imputation_ &varname RowPercent RowStdErr);
   set temptab;
   if young = 1 and &varname in (1,2,3,4);
run;

proc sort data = temptab2;
   by &varname _imputation_;
run;

proc mianalyze data = temptab2;
   by &varname;
   modeleffects RowPercent;
   stderr RowStdErr;
run;
%mend miest;

%miest(py_tob_chg);

/* Perform imputations based on PMMs for the use variables with missing data, and consider
alternative adjustments for non-ignorability */
```

```
/* NOTE: There are many possible choices for the shift parameters applied to each generalized
logit function! These selections were based on Tables 2 and 3,
and the selection models. */

proc mi data=desk.use_forimp seed=41279 nimpute=10 out=outfcs_discrim3;
   class sex ethrace2a educate income2 smoker consumer ltsmoker W2CONSUMER w1mj w2mj;
   fcs logistic (ltsmoker W2CONSUMER / link=glogit);
   fcs logistic (w1mj w2mj);
   mnar adjust (ltsmoker (event = '1') / shift = 0.5);
   *mnar adjust (ltsmoker (event = '2') / shift = -0.5);
   *mnar adjust (ltsmoker (event = '3') / shift = -0.5);
   mnar adjust (ltsmoker (event = '4') / shift = 0.5);
   mnar adjust (W2CONSUMER (event = '1') / shift = -0.5);
   mnar adjust (W2CONSUMER (event = '2') / shift = 0.5);
   mnar adjust (W2CONSUMER (event = '3') / shift = 0.5); /* NR more likely to be cat 2 and 3 */
   mnar adjust (w2mj (event = '1') / shift = 0.5);
   var ltsmoker W2CONSUMER w1mj w2mj age sex ethrace2a educate
      income2 smoker consumer any_py_anxd any_py_moodd any_lt_persd;
run;

/* Compute values on use change variables based on PMM imputations */

data outfcs_discrim4;
   set outfcs_discrim3;
   if (smoker = 2 or smoker = 3) and (ltsmoker = 2 or ltsmoker = 3 or ltsmoker = 4) then
py_tob_chg = 1;
   else if (smoker = 1) and (ltsmoker = 2 or ltsmoker = 3 or ltsmoker = 4) then py_tob_chg = 2;
   else if (smoker = 2 or smoker = 3) and (ltsmoker = 1) then py_tob_chg = 3;
   else if smoker = 1 and ltsmoker = 1 then py tob chg = 4;
   if (consumer = 2 or consumer = 3) and (W2CONSUMER = 2 or W2CONSUMER = 3) then py_alc_chg = 1;
   else if (consumer = 1) and (W2CONSUMER = 2 or W2CONSUMER = 3) then py_alc_chg = 2;
   else if (consumer = 2 or consumer = 3) and (W2CONSUMER = 1) then py_alc_chg = 3;
   else if consumer = 1 and W2CONSUMER = 1 then py_alc_chg = 4;
   if w1mj = 0 and w2mj = 0 then py_mj_chg2 = 1;
   else if w1mj = 1 and w2mj = 0 then py mj chg2 = 2;
   else if w1mj = 0 and w2mj = 1 then py_mj_chg2 = 3;
   else if w1mj = 1 and w2mj = 1 then py_mj_chg2 = 4;
   if (18 <= age <= 24) then young = 1;
   else young = 0;
run;

/* Perform MI analysis for a given variable (Table 4) */

%macro PMMmiest(varname);
proc surveyfreq data=outfcs_discrim4;
   weight weight;
   cluster psu;
   stratum stratum;
   tables _imputation_*young*&varname / row;
   ods output CrossTabs = temptab;
run;

data temptab2 (keep = _Imputation_ &varname RowPercent RowStdErr);
   set temptab;
   if young = 1 and &varname in (1,2,3,4);
   *if _Imputation_ not in (2,5,10); /* for py_tob_chg, throw out imputations where models did
not converge */
run;

proc sort data = temptab2;
   by &varname _imputation_;
run;

proc mianalyze data = temptab2;
   by &varname;
   modeleffects RowPercent;
   stderr RowStdErr;
run;
%mend PMMmiest;

%PMMmiest(py_tob_chg);
```

```
%PMMmiest(py_alc_chg);
%PMMmiest(py_mj_chg2);

/*****************************************************************************************
*****************************************/
/* Perform imputations based on PMMs for the disorder variables with missing data, and consider
alternative adjustments for non-ignorability */
/*****************************************************************************************
*****************************************/

/* NOTE: shift choices are based on predictors of attrition at Wave 2 and selection model
results; other choices are certainly possible! */

proc mi data=desk.dis_forimp seed=41279 nimpute=10 out=outfcs_discrim5;
   class sex ethrace2a educate income2 TAB12MDX ALCABDEP12DX curtabdep W2AAD12 mjud2 mjud1;
   fcs logistic (curtabdep W2AAD12 mjud2 / link=glogit);
   mnar adjust (curtabdep (event = '1') / shift = 0.5);
   mnar adjust (W2AAD12 (event = '0') / shift = -0.5);
   mnar adjust (W2AAD12 (event = '1') / shift = 0.5);
   mnar adjust (W2AAD12 (event = '2') / shift = 0.5);
   mnar adjust (W2AAD12 (event = '3') / shift = 0.5);
   mnar adjust (mjud2 (event = '1') / shift = 0.5);
   var curtabdep W2AAD12 mjud2 mjud1 age sex ethrace2a educate
      income2 TAB12MDX ALCABDEP12DX any_py_anxd any_py_moodd any_lt_persd;
run;

/* Compute values on disorder change variables based on PMM imputations */

data outfcs_discrim6;
   set outfcs_discrim5;
   if TAB12MDX = 0 and curtabdep = 0 then py_nicd_chg = 1;
   else if TAB12MDX = 1 and curtabdep = 0 then py_nicd_chg = 2;
   else if TAB12MDX = 0 and curtabdep = 1 then py_nicd_chg = 3;
   else if TAB12MDX = 1 and curtabdep = 1 then py_nicd_chg = 4;
   if ALCABDEP12DX = 0 and W2AAD12 = 0 then py_alcd_chg = 1;
   else if ALCABDEP12DX >= 1 and ALCABDEP12DX <= 3 and W2AAD12 = 0 then py_alcd_chg = 2;
   else if ALCABDEP12DX = 0 and W2AAD12 >= 1 and W2AAD12 <= 3 then py_alcd_chg = 3;
   else if ALCABDEP12DX >= 1 and ALCABDEP12DX <= 3 and W2AAD12 >= 1 and W2AAD12 <= 3 then
py_alcd_chg = 4;
   if mjud1 = 0 and mjud2 = 0 then py_mjud_chg2 = 1;
   else if mjud1 = 1 and mjud2 = 0 then py_mjud_chg2 = 2;
   else if mjud1 = 0 and mjud2 = 1 then py_mjud_chg2 = 3;
   else if mjud1 = 1 and mjud2 = 1 then py_mjud_chg2 = 4;
   if (18 <= age <= 24) then young = 1;
   else young = 0;
run;

/* Perform MI analysis for a given variable (Table 4) */

%macro PMMmiest(varname);
proc surveyfreq data=outfcs_discrim6;
   weight weight;
   cluster psu;
   stratum stratum;
   tables _imputation_*young*&varname / row;
   ods output CrossTabs = temptab;
run;

data temptab2 (keep = _Imputation_ &varname RowPercent RowStdErr);
   set temptab;
   if young = 1 and &varname in (1,2,3,4);
run;

proc sort data = temptab2;
   by &varname _imputation_;
run;

proc mianalyze data = temptab2;
   by &varname;
   modeleffects RowPercent;
   stderr RowStdErr;
```

```
run;
%mend PMMmiest;

%PMMmiest(py_nicd_chg);
%PMMmiest(py_alcd_chg);
%PMMmiest(py_mjud_chg2);
```