```
/* Sentinel Lab-Based ADR Survey */
/* Last updated: 8/3/2022 */
/* Uses STATA version 16.1 */

/* This code assumes participant- and laboratory-level information follow the
configuration of the Excel data upload template and are stored in a data file
labeled "patient_data_sentinel.xlsx". */
/* HIV drug resistance sequences are assumed to be in FASTA file format and stored
in a data file labeled "FASTA_sentinel.xlsx". Survey identification numbers must
follow WHO convention. */
/* Both data files should be in the same directory, e.g., "C:/Documents" */


/* Clear any previous output */
clear

/* Set working directory to the directory containing all data files. For example, if directory is "C:/Documents", run the following code */
cd "C:/Documents"

/* Import each sheet of the Excel data capture tool, storing the first row as headings
and changing all header names to uppercase. Save each sheet as its own .dta file.
Given the Excel file name "patient_data_sentinel.xlsx", run the following*/
```

```
following*/
import excel using "patient_data_sentinel.xlsx", describe
forvalues sheet=1/`=r(N_worksheet)' {
    local sheetname = r(worksheet_`sheet')
    import excel using patient_data_sentinel, sheet("`sheetname'")
firstrow case(upper)
    local sheetname = upper(subinstr("`sheetname'", " ", "_", .))
    save "`sheetname'", replace
    clear
}


/* This code prepares the resistance data imported from the
Stanford HIVdb database */

/* Import the HIVdb resistance data, storing the first row as
heading and
changing all header names to uppercase. Given the file name
"FASTA_sentinel.xlsx", run */
import excel using "FASTA_sentinel.xlsx", sheet("ResistanceSummary"
) firstrow case(upper)
/* Rename SEQUENCENAME as PARTICIPANTID */
rename SEQUENCENAME PARTICIPANTID
/* Drop all cells without a subject ID */
drop if missing(PARTICIPANTID)
/* Drop all unnecessary variables */
drop *SCORE ALGORITHM* STRAIN GENES PI* NRTI* NNRTI* INSTI*
/* Replace NA's with the missing symbol */
destring, ignore("NA") replace
/* For each of the resistance level variables, classify as a
binary resistance
indicator, with levels 1-2 corresponding to susceptible (no
HIVDR), and levels
```

```
48  indicator, with levels 1-2 corresponding to susceptible (no
    HIVDR), and levels
49  3-5 corresponding to HIVDR. */
50  ds *LEVEL
51  local plist = r(varlist)
52  foreach i of local plist {
53      replace `i' = 0 if `i' < 3 & !missing(`i')
54      replace `i' = 1 if `i' >= 3 & !missing(`i')
55  }
56  /* Rename resistance type variables */
57  rename *LEVEL *_RES
58  /* Generate variable for DTG-specific resistance */
59  gen DTG_ADR = DTG_RES
60  /* save changes to the modified dataset */
61  save RESISTANCE_SUMMARY, replace
62
63
64  /* This code prepares the VL laboratory data */
65  /* Remove previous dataset, then load the VL lab data and rename
    the variables */
66  clear
67  use VL_LAB_INFORMATION.dta
68  rename NAME* LABNAME
69  rename SITECODE* LABCODE
70  rename PROP* PROPWINDOW
71  /* Exclude observations missing a lab code */
72  drop if missing(LABCODE)
73  /* Save changes */
74  save VL_LAB_INFORMATION, replace
75
76
77  /* This code prepares the patient-level data on ARV treatment
    regimen */
78
```

```
78
79  /* Remove previous dataset, then load the treatment regimen data */
80  clear
81  use PARTICIPANT_TREATMENTS.dta
82  /* Exclude observations missing a subject ID or corresponding to
    past ART */
83  drop if missing(PARTICIPANTID) | upper(CURRENTARTYN) == "N"
84  /* Drop unnecessary variables */
85  drop OTHERARVDRUG CURRENTARTYN
86  /* Rename ARV drug types so that all variable names start with a
    letter */
87  replace ARVDRUG = "ARV_" + ARVDRUG
88  /* Generate binary DTG variable corresponding to 1 if patient is
    on a
89  DTG-containing regimen and 0 if patient is on a non-DTG-containing
    regimen */
90  gen TEMP_DTG = cond(inlist(ARVDRUG, "ARV_DTG", "ARV_TLD", "ARV_JUL"
    ), 1, 0)
91  by PARTICIPANTID, sort: egen DTG = max(TEMP_DTG)
92  drop TEMP_DTG
93  /* Reformat the ARVDRUG variable so that each ARV drug type
    corresponds to a
94  new binary variable, set to 1 if the patient is taking that drug,
    and set to 0
95  if not */
96  gen ON = 1
97  reshape wide ON, i(PARTICIPANTID) j(ARVDRUG) string
98  rename ON* *
99  /* Save changes */
100 save PARTICIPANT_TREATMENTS, replace
101
```

```
101
102
103 /* This code prepares the patient-level data on other variables */
104
105 /* Remove previous dataset, then load the patient-level data */
106 clear
107 use SURVEY_PARTICIPANTS.dta
108 /* Drop if missing subject ID or patient is not in the adult
    population */
109 /* If analysis is for children and adolescents instead, uncomment
    the following
110 line of code and run it instead */
111 /* drop if missing(PARTICIPANTID) | substr(PARTICIPANTID, -1, .)
    != "c"
112 */
113 drop if missing(PARTICIPANTID) | substr(PARTICIPANTID, -1, .) != "a"
114 /* Rename variable corresponding to the VL laboratory for each
    patient */
115 rename SITECODE* LABCODE
116 rename DATEOFINITIATION* DATEINIT
117 rename AGE* AGE
118 rename DATEOFBIRTH* DATEOFBIRTH
119 rename GENDER* GENDER
120 rename LABSPEC* LABSPECIMENCODE
121 /* Recode unknown values as the missing symbol */
122 recode DATE* (9999 = .)
123 recode AGE (-9 = .)
124 /* Save changes */
125 save SURVEY_PARTICIPANTS.dta, replace
126
127
```

```
127
128 /* Merge all the datasets and save the combined data into one .dta
    file */
129
130 /* Use a many-to-one merge to merge the VL laboratory data */
131 merge m:1 LABCODE using VL_LAB_INFORMATION, keep(match) nogenerate
132 /* Merge in the treatment regimen data by subject ID */
133 merge 1:1 PARTICIPANTID using PARTICIPANT_TREATMENTS, keep(match)
    nogenerate
134 /* Merge in the HIVDR data by subject ID */
135 merge 1:1 PARTICIPANTID using RESISTANCE_SUMMARY, keep(match)
    nogenerate
136 save ALL_DATA.dta, replace
137
138
139 /* This code is for creating the finite populations and other
    necessary
140 variables for survey design */
141
142 /* Remove previous dataset and load in the combined data */
143 clear
144 use ALL_DATA.dta
145 /* Generate the variable for the total eligible specimens forming
    the finite
146 population for each laboratory. This is equal to n_sampled /
    prop_window * 4 */
147 bysort LABCODE: gen FINITE_POP = _N / PROPWINDOW * 4
148 /* Generate the stratum weights, calculated as the stratum totals
    divided by
149 the number of sampled case specimens per stratum */
```