# aPpendix i: sample codes for icd diagnosis codes

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

XXXX.SAS -- ICD-10 diagnosis codes for medical condition XXXX

This program are including

1) dataset to input ICD-10 diagnosis codes for medical condition XXXX.

2) macro variables used in different SAS statements

3) macros to match varied databases and coding methods

Instructions:

XXXX is the given name of a medical condition. It also matches the

title of this program

Example:

DZNAME replaces XXXX in this paper

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/;

/\*Input ICD-10 Diagnosis Code\*/

Data ICD10;

length ICDCode $ 6;

input ICDCode $ lag;

datalines;

;

run;

/\*Get Macro Variables\*/

data ICD10\_all;

set ICD10;

retain order;

order+1;

length newicd $32;

if order=1 then do;

if lag in (1 2) then newicd='&coding'||" =: '"||cats(icdcode)||"'";

else if lag=0 then newicd='&coding'||" = '"||cats(icdcode)||"'";

end;

else do;

if lag in (1 2) then newicd='or &coding'||" =: '"||cats(icdcode)||"'";

else if lag=0 then newicd='or &coding'||" = '"||cats(icdcode)||"'";

end;

run;

/\*Macro variable used for none pass-thru coding in CDW\*/

proc sql noprint;

select newicd

into :XXXX10 separated by ' '

from work.ICD10\_All;

quit;

data ICD10\_SQL;

set ICD10;

retain order;

order+1;

length newicd $32;

if order=1 then do;

if lag in (1 2) then newicd='&coding'||" like '"||cats(icdcode)||"%'";

else if lag=0 then newicd='&coding'||" = '"||cats(icdcode)||"'";

end;

else do;

if lag in (1 2) then newicd='or &coding'||" like '"||cats(icdcode)||"%'";

else if lag=0 then newicd='or &coding'||" = '"||cats(icdcode)||"'";

end;

run;

/\*Macro variable used for explicity pass-thru coding method in CDW\*/

proc sql noprint;

select newicd

into :XXXX10\_SQL separated by ' '

from work.ICD10\_SQL;

quit;

data ICD10\_OMOP;

set ICD10;

retain order;

order+1;

length newicd $32;

if order=1 then do;

if lag in (1 2) then newicd='&coding'||" like 'ICD10|"||cats(icdcode)||"%'";

else if lag=0 then newicd='&coding'||" = 'ICD10|"||cats(icdcode)||"'";

end;

else do;

if lag in (1 2) then newicd='or &coding'||" like 'ICD10|"||cats(icdcode)||"%'";

else if lag=0 then newicd='or &coding'||" = 'ICD10|"||cats(icdcode)||"'";

end;

run;

/\*Macro variable used for OMOP data with explicity pass-thru coding\*/

proc sql noprint;

select newicd

into :XXXX10\_OMOP separated by ' '

from work.ICD10\_OMOP;

quit;

data ICD10\_CMS;

set ICD10;

retain order;

order+1;

length newicd $32;

if order=1 then do;

if Lag=1 then newicd='&coding'||"=substr("||cats(compress(icdcode, '.'))||", 1, 3)";

else if lag=2 then newicd='&coding'||"=substr("||cats(compress(icdcode, '.'))||", 1, 4)";

else if lag=0 then newicd='&coding'||"= '"||cats(compress(icdcode, '.'))||"'";

end;

else do;

if Lag=1 then newicd='or &coding'||"=substr("||cats(compress(icdcode, '.'))||", 1, 3)";

else if lag=2 then newicd='&coding'||"=substr("||cats(compress(icdcode, '.'))||", 1, 4)";

else if lag=0 then newicd='or &coding'||"= '"||cats(compress(icdcode, '.'))||"'";

end;

run;

/\*Macro variable used for CMS files\*/

proc sql noprint;

select newicd

into :XXXX10\_CMS separated by ' '

from work.ICD10\_CMS;

quit;

proc datasets;

delete ICD10 ICD10\_All ICD10\_SQL ICD10\_OMOP ICD10\_CMS;

quit;

/\*Macro used in SAS statement during data step\*/

%macro XXXX\_ICD10 (coding=);

if &XXXX10 then do;

&outvar='XXXX';

&outvar.\_label='Label\_Text';

end;

%mend XXXX\_ICD10;

/\*Macro used in SAS statement for explicity pass-thru coidng\*/

%macro XXXX\_ICD10\_SQL (coding=);

(&XXXX10\_SQL)

%mend XXXX\_ICD10\_SQL;

/\*Macro used in SAS statement for OMOP by explicity pass-thru coidng\*/

%macro XXXX\_ICD10\_OMOP (coding=);

(&XXXX10\_OMOP)

%mend XXXX\_ICD10\_OMOP;

/\*Macro used in SAS statement for CMS files\*/

%macro XXXX\_ICD10\_CMS (coding=);

if i=1 then do;

if &XXXX10\_CMS

then do;

&outvar.='XXXX';

&outvar.\_label='Label\_Text';

&outvar.\_primary=1;

&outvar.\_code=&coding;

stop=1;

end;

else stop=0;

end;

else do;

if &XXXX10\_CMS

then do;

&outvar.='XXXX';

&outvar.\_label='Label\_Text';

&outvar.\_primary=2;

&outvar.\_code=&coding;

stop=1;

end;

else stop=0;

end;

%mend XXXX\_ICD10\_CMS;

# appendix iI: sample codes for procedure codes (cpt)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

XXXX.SAS -- CPT codes for surgical procedure XXXX

This program are including

1) dataset to input CPT codes for surgical procedure XXXX.

2) macro variables used in different SAS statements

3) macros to match varied databases and coding methods

Instructions:

XXXX is the given name of a medical condition. It also matches the

title of this program

Example:

PROCNAME1 replaces XXXX in this paper

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/;

/\*Input CPT Code\*/

Data CPT;

length CPT\_Code $ 6;

input CPT\_Code $ lag;

datalines;

;

run;

/\*Get Macro Variables\*/

data CPT\_all;

set CPT;

retain order;

order+1;

length NewCPT $32;

if order=1 then do;

if Lag in (1 2) then NewCPT='&coding'||"=substr("||cats(CPT\_Code)||", 1, 3)";

else if lag=0 then NewCPT='&coding'||" = '"||cats(CPT\_Code)||"'";

end;

else do;

if Lag in (1 2) then NewCPT='&coding'||"=substr("||cats(CPT\_Code)||", 1, 3)";

else if lag=0 then NewCPT='or &coding'||" = '"||cats(CPT\_Code)||"'";

end;

run;

/\*Macro variable used for none pass-thru coding in CDW\*/

proc sql noprint;

select NewCPT

into :XXXX\_CPT separated by ' '

from work.CPT\_All;

quit;

data CPT\_SQL;

set CPT;

retain order;

order+1;

length NewCPT $32;

if order=1 then do;

if Lag in (1 2) then NewCPT='&coding'||" like '"||cats(CPT\_Code)||"%'";

else if lag=0 then NewCPT='&coding'||" = '"||cats(CPT\_Code)||"'";

end;

else do;

if Lag in (1 2) then NewCPT='or &coding'||" like '"||cats(CPT\_Code)||"%'";

else if lag=0 then NewCPT='or &coding'||" = '"||cats(CPT\_Code)||"'";

end;

run;

/\*Macro variable used for explicity pass-thru coding method in CDW\*/

proc sql noprint;

select NewCPT

into :XXXX\_CPT\_SQL separated by ' '

from work.CPT\_SQL;

quit;

data CPT\_OMOP;

set CPT;

retain order;

order+1;

length NewCPT $32;

if order=1 then do;

if Lag in (1 2) then NewCPT='&coding'||" like 'CPT|"||cats(CPT\_Code)||"%'";

else if lag=0 then NewCPT='&coding'||" = 'CPT|"||cats(CPT\_Code)||"'";

end;

else do;

if Lag in (1 2) then NewCPT='or &coding'||" like 'CPT|"||cats(CPT\_Code)||"%'";

else if lag=0 then NewCPT='or &coding'||" = 'CPT|"||cats(CPT\_Code)||"'";

end;

run;

/\*Macro variable used for OMOP data with explicity pass-thru coding\*/

proc sql noprint;

select NewCPT

into :XXXX\_CPT\_OMOP separated by ' '

from work.CPT\_OMOP;

quit;

data CPT\_CMS;

set CPT;

retain order;

order+1;

length NewCPT $32;

if order=1 then do;

if Lag=1 then NewCPT='&coding'||"=substr("||cats(CPT\_Code)||", 1, 3)";

else if Lag=2 then NewCPT='&coding'||"=substr("||cats(CPT\_Code)||", 1, 4)";

else if lag=0 then NewCPT='&coding'||" = '"||cats(CPT\_Code)||"'";

end;

else do;

if Lag=1 then NewCPT='&coding'||"=substr("||cats(CPT\_Code)||", 1, 3)";

else if Lag=2 then NewCPT='&coding'||"=substr("||cats(CPT\_Code)||", 1, 4)";

else if lag=0 then NewCPT='or &coding'||" = '"||cats(CPT\_Code)||"'";

end;

run;

/\*Macro variable used for CMS files\*/

proc sql noprint;

select NewCPT

into :XXXX\_CPT\_CMS separated by ' '

from work.CPT\_CMS;

quit;

proc datasets;

delete CPT CPT\_All CPT\_SQL CPT\_OMOP CPT\_CMS;

quit;

/\*Macro used in SAS statement during data step\*/

%macro XXXX\_CPT (coding=);

if &XXXX\_CPT then do;

&outvar='XXXX';

&outvar.\_label='Text\_XXX';

end;

%mend XXXX\_CPT;

/\*Macro used in SAS statement for explicity pass-thru coidng\*/

%macro XXXX\_CPT\_SQL (coding=);

(&XXXX\_CPT\_SQL)

%mend XXXX\_CPT\_SQL;

/\*Macro used in SAS statement for OMOP by explicity pass-thru coidng\*/

%macro XXXX\_CPT\_OMOP (coding=);

(&XXXX\_CPT\_OMOP)

%mend XXXX\_CPT\_OMOP;

/\*Macro used in SAS statement for CMS files\*/

%macro XXXX\_CPT\_CMS (coding=);

if i=1 then do;

if &XXXX\_CPT\_CMS

then do;

&outvar.='XXXX';

&outvar.\_label='Text\_XXX';

&outvar.\_primary=1;

&outvar.\_code=&coding;

stop=1;

end;

else stop=0;

end;

else do;

if &XXXX\_CPT\_CMS

then do;

&outvar.='XXXX';

&outvar.\_label='Text\_XXX';

&outvar.\_primary=2;

&outvar.\_code=&coding;

stop=1;

end;

else stop=0;

end;

%mend XXXX\_CPT\_CMS;

# appendix iII longitudinal\_data\_macro.sas

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Longitudinal\_Data\_Macro.sas -- macro to assemble individual dataset to

one longitudinal data structure

Instructions:

The macro arguments are as follows, where "?" indicates a

required argument:

? inputdata = all input datasets extracted from database

? dizname\_all = names of all included medical conditions and procedures.

? dizname\_ICD10 = names of all included medical conditions,

use XXX instead none.

? dizname\_PRO10 = names of all included procedures if have ICD procedure code,

use XXX instead none.

? dizname\_CPT = names of all included procedures if have CPT code,

use XXX instead none.

? dizname\_STOP = names of all included procedures if have VA Stop code,

use XXX instead none.

? coding = original variable name in extracted datasets.

? outvar = variable name is going to used in longitudinal dataset.

Example:

%Let inputdata=Outpatients Inpatients Procedures;

%let dizname\_all=DZNAME DZNAME1 DZNAME2 PROCNAME1 PROCNAME2 PROCNAME3;

%let dizname\_ICD10=DZNAME DZNAME1 DZNAME2;

%let dizname\_PRO10=PROCNAME1 PROCNAME2;

%let dizname\_CPT=PROCNAME1 PROCNAME2 PROCNAME3;

%let dizname\_STOP=XXX;

%let coding=ICDCode;

%let outvar=Comorbidities;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/;

%macro getdataset;

data \_null\_;

array \_ndata &inputdata;

call symput('n\_data', left(put(dim(\_ndata), 2.0)));

run;

data indata;

set

%do h=1 %to &n\_data;

%let orgdata= %scan(&inputdata, &h);

&orgdata

%end;

;

run;

%mend getdataset;

%macro Longdata\_cdw;

data \_null\_;

array \_xvars &dizname\_all;

call symput('n\_all', left(put(dim(\_xvars), 2.0)));

array \_cvars &dizname\_ICD10;

call symput('n\_icd10', left(put(dim(\_cvars), 2.0)));

array \_dvars &dizname\_ICD9;

call symput('n\_icd9', left(put(dim(\_dvars), 2.0)));

array \_evars &dizname\_PRO10;

call symput('n\_pro10', left(put(dim(\_evars), 2.0)));

array \_fvars &dizname\_PRO9;

call symput('n\_pro9', left(put(dim(\_fvars), 2.0)));

array \_gvars &dizname\_CPT;

call symput('n\_cpt', left(put(dim(\_gvars), 2.0)));

array \_hvars &dizname\_stop;

call symput('n\_stop', left(put(dim(\_hvars), 2.0)));

run;

%put number &n\_all &n\_icd10 &n\_icd9 &n\_pro10 &n\_pro9 &n\_cpt &n\_data;

%do m=1 %to &n\_all;

%let xi= %scan(&dizname\_all, &m);

%let xi\_text=%lowcase(&xi);

%put name &xi;

/\*setup default setting\*/

%let icd10=no;

%let icd9=no;

%let procedure9=no;

%let procedure10=no;

%let cpt=no;

%let stopcode=no;

/\*identify icd10 diagnosise code\*/;

%do c = 1 %to &n\_icd10;

%if %lowcase(&xi) = %lowcase(%scan(&dizname\_ICD10, &c)) %then %do;

%let icd10 = yes;

%end;

%end;

%put ICD10=&icd10;

/\*identify icd9 diagnosise code\*/;

%do d = 1 %to &n\_icd9;

%if %lowcase(&xi) = %lowcase(%scan(&dizname\_ICD9, &d)) %then %do;

%let icd9 = yes;

%end;

%end;

%put ICD9=&icd9;

/\*identify icd10 procedure code\*/;

%do e = 1 %to &n\_pro10;

%if %lowcase(&xi) = %lowcase(%scan(&dizname\_pro10, &e)) %then %do;

%let procedure10 = yes;

%end;

%end;

%put ICD\_Procedure10=&procedure10;

/\*identify icd9 procedure code\*/;

%do f = 1 %to &n\_pro9;

%if %lowcase(&xi) = %lowcase(%scan(&dizname\_pro9, &f)) %then %do;

%let procedure9 = yes;

%end;

%end;

%put ICD\_Procedure9=&procedure9;

/\*identify CPT/HCPCS code\*/;

%do g = 1 %to &n\_cpt;

%if %lowcase(&xi) = %lowcase(%scan(&dizname\_cpt, &g)) %then %do;

%let cpt = yes;

%end;

%end;

%put CPT=&cpt;

/\*identify stop code\*/;

%do h = 1 %to &n\_stop;

%if %lowcase(&xi) = %lowcase(%scan(&dizname\_stop, &h)) %then %do;

%let stopcode = yes;

%end;

%end;

%put STOPCODE=&stopcode;

/\*setup null dataset\*/

data &xi.\_temp; set \_null\_; run;

%if &icd10=yes %then %do;

data &xi.\_icd10;

set indata;

length &outvar $32 &outvar.\_label $50;

%&xi.\_icd10(coding=&coding);

keep subjectid dates &coding &outvar &outvar.\_label;

run;

data &xi.\_temp;

set &xi.\_temp

&xi.\_icd10 (where=(lowcase(&outvar)="&xi\_text"))

;

run;

%end;

%if &icd9=yes %then %do;

data &xi.\_icd9;

set indata;

length &outvar $32 &outvar.\_label $50;

%&xi.\_icd9(coding=&coding);

keep subjectid dates &coding &outvar &outvar.\_label;

run;

data &xi.\_temp;

set &xi.\_temp

&xi.\_icd9 (where=(lowcase(&outvar)="&xi\_text"))

;

run;

%end;

%if &procedure10=yes %then %do;

data &xi.\_pro10;

set indata;

length &outvar $32 &outvar.\_label $50;

%&xi.\_procedure10(coding=&coding);

keep subjectid dates &coding &outvar &outvar.\_label;

run;

data &xi.\_temp;

set &xi.\_temp

&xi.\_pro10 (where=(lowcase(&outvar)="&xi\_text"))

;

run;

%end;

%if &procedure9=yes %then %do;

data &xi.\_pro9;

set indata;

length &outvar $32 &outvar.\_label $50;

%&xi.\_procedure9(coding=&coding);

keep subjectid dates &coding &outvar &outvar.\_label;

run;

data &xi.\_temp;

set &xi.\_temp

&xi.\_pro9 (where=(lowcase(&outvar)="&xi\_text"))

;

run;

%end;

%if &cpt=yes %then %do;

data &xi.\_cpt;

set indata;

length &outvar $32 &outvar.\_label $50;

%&xi.\_cpt(coding=&coding);

keep subjectid dates &coding &outvar &outvar.\_label;

run;

data &xi.\_temp;

set &xi.\_temp

&xi.\_cpt (where=(lowcase(&outvar)="&xi\_text"))

;

run;

%end;

%if &stopcode=yes %then %do;

data &xi.\_stop;

set indata;

length &outvar $32 &outvar.\_label $50;

%&xi.\_stop(coding=&coding);

keep subjectid dates &coding &outvar &outvar.\_label;

run;

data &xi.\_temp;

set &xi.\_temp

&xi.\_stop (where=(lowcase(&outvar)="&xi\_text"))

;

run;

%end;

proc sort data=&xi.\_temp nodupkey; by subjectid dates &coding; run;

data report;

set report

&xi.\_temp

;

run;

proc datasets; delete &xi.\_icd10 &xi.\_icd9 &xi.\_pro10 &xi.\_pro9 &xi.\_cpt &xi.\_stop

&xi.\_temp; quit;

%end;

%mend Longdata\_CDW;

# appendix iV observation\_data\_macro.sas

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Observation\_Data\_Macro.sas -- macro to convert longitudinal data structure to

analytical dataset (one observation one subject)

Instructions:

The macro arguments are as follows, where "?" indicates a

required argument:

? outvar = variable name used in longitudinal dataset.

? in\_out = output dataset.

? casenum = number of enconter.

? varname = names of all included medical conditions and procedures.

Example:

%let outvar=Comorbidities;

%let in\_out=Obs\_data;

%let casenum=1;

%let varname=DZNAME DZNAME1 DZNAME2 PROCNAME1 PROCNAME2 PROCNAME3;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/;

%macro Obs\_CDW (Indata=);

data \_null\_;

array \_xvars &varname;

call symput('n\_names', left(put(dim(\_xvars), 2.0)));

run;

%put &n\_names;

%do m = 1 %to &n\_names;

%let xi = %scan(&varname, &m);

%let xi\_text=%lowcase(&xi);

%put working on predictor variable &m &xi &xi\_text;

data &xi.\_temp;

set &indata

;

by subjectid;

where lowcase(&outvar)="&xi\_text";

&xi.=1;

call symput("labelname", &outvar.\_label);

proc sort nodupkey; by subjectid dates;

run;

data &xi.\_nodup;

set &xi.\_temp

;

by subjectid;

if first.subjectid;

keep subjectid &xi;

run;

data &xi.\_first;

set &xi.\_temp;

by subjectid dates;

if first.subjectid=1;

rename dates=fstdates\_&xi;

keep subjectid dates;

run;

data &xi.\_last;

set &xi.\_temp ;

by subjectid;

keep subjectid dates count;

rename dates=lastdates\_&xi;

if first.subjectid then count=0;

count+1;

if last.subjectid then output;

run;

data &xi.\_&in\_out;

merge &xi.\_nodup &xi.\_first &xi.\_last(in=inlast where=(count>=&casenum));

by subjectid;

if inlast;

drop count;

label &xi="&labelname";

run;

proc datasets; delete &xi.\_temp &xi.\_nodup &xi.\_first &xi.\_last; quit;

%end;

%mend Obs\_CDW;

%macro obs\_data;

data \_null\_;

array \_xvars &varname;

call symput('n\_names', left(put(dim(\_xvars), 2.0)));

array \_svars &datasets;

call symput('n\_sets', left(put(dim(\_svars), 2.0)));

run;

%put &n\_names;

%do m = 1 %to &n\_names;

%let xi = %scan(&varname, &m);

%put working on predictor variable &m &xi;

data &xi.\_all;

set

%do h=1 %to &n\_sets;

%let inputset= %scan(&datasets, &h);

&xi.\_&inputset

%end;

;

where &xi=1;

proc sort nodupkey; by subjectid fstdates\_&xi;

proc sort nodupkey; by subjectid;

run;

data report;

merge report

&xi.\_all

;

by subjectid;

run;

proc datasets; delete &xi.\_all; quit;

%end;

%mend obs\_data;