PROC IMPORT DATAFILE= "C:\Users\vw233\OneDrive - Yale University\Desktop

\Ryan White\RW71223.xlsx"

OUT= HIV

dbms = XLSX

REPLACE;

GETNAMES=YES;

sheet ="Public Dataset no Clinic Survey";

options MSGLEVEL = i;

RUN;

proc format;

value HIV\_CriminalizationCode

 1 = 'HIV Laws' 2= 'Sentence Enhancement' 3= 'General Crime Sentencing' 4='HIV Law and Gen Crime Sentencing' 5='Sentence Enhancement and General Crime Sentencing' 6='None';

value Urban\_Suburban\_ruralCode

1='Urban' 2='Suburban' 3='rural';

value UrbanicityCode

1 ='Large Urban' 2='Large Fringe Urban' 3='Medium Urban' 4='Small Urban' 5='Micropolitan' 6='Non-core';

value Water\_ViolationsCode

1 ='No' 2='Yes';

run;

Data Work.Hiv;

Set Work.Hiv;

Format HIV\_Criminalization HIV\_CriminalizationCode. Urban\_Suburban\_rural Urban\_Suburban\_ruralCode. Urbanicity UrbanicityCode. Water\_Violations Water\_ViolationsCode.;

Run;

DATA work.Hiv;

SET work.Hiv;

\_1MRC=\_1\_More\_Race\_Center \*100;

\_100PLC =\_100\_Poverty\_Line\_Center\*100;

\_1324HIV = \_13\_24\_HIV\*100;

\_1324NH = \_13\_24\_New\_HIV \* 100;

\_200PLC=\_200\_Poverty\_Line\_Center\*100;

\_2534HIV=\_25\_34\_HIV\*100;

\_2534NH=\_25\_34\_New\_HIV\*100;

\_3544H=\_35\_44\_HIV\*100;

\_3544NH=\_35\_44\_New\_HIV\*100;

\_4554H=\_45\_54\_HIV\*100;

\_4554NH=\_45\_54\_New\_HIV\*100;

\_55PH=\_55PLUS\_HIV\*100;

\_55PNH=\_55PLUS\_NEW\_HIV\*100;

AC=Access\_to\_PN\_Care\*100;

AHF=AccessHealthyFood\*100;

AWC=AdolWeight\_Center\*100;

ACE=Adult\_Center\*100;

AWSC=Adult\_Weight\_Screening\_Center\*100;

ADD=Alcohol\_Driving\_Death\*100;

AI=American\_Indian\*100;

AMS=Annual\_Mammo\_Screen\*100;

As=Asian\*100;

AsC=Asian\_Center\*100;

AsthC=Asthma\_Center\*100;

APRP=AIDS\_Prev\_Rate/1000;

ATC=Asthma\_Treatment\_Center\*100;

BH=Bachelors\_Higher\*100;

BPL=BelowPovHouse\_lvl\*100;

BL=Black\*100;

BAAC=Black\_African\_American\_Center\*100;

BHIV=Black\_HIV\*100;

BPCC=BP\_Control\_Center\*100;

BPUCC=100-BPCC;

BRCC=BrCancer\_Center\*100;

BBA=Broadband\_access\*100;

CSC=Cervical\_Screening\_Center\*100;

CCC=Child\_Care\_Cost\*100;

CVC=Childhood\_Vac\_Center\*100;

Chil=Children\*100;

ChilC=Children\_Center\*100;

CFRL=Children\_Free\_Reduced\_lunch\*100;

CIP=Children\_In\_Poverty\*100;

CSP=Children\_Single\_Parent\*100;

CRC=CRC\_Screening\_Center\*100;

DeC=Dental\_Center\*100;

DSC=Dental\_Sealants\_Center\*100;

DRC=Depression\_Rem\_Center\*100;

DSCC=Depression\_Screening\_Center\*100;

DC=Diabetes\_Center\*100;

DP=Diabetes\_Prevalence\*100;

D65=Disability\_Under65\*100;

DIS=Disabled\*100;

DY=Disconnected\_Youth\*100;

DE=DrinkingExcess\*100;

DAW=Driving\_Alone\_Work\*100;

EM=Employment\*100;

ENC=Enabling\_Center\*100;

EXO=Exercise\_opp\*100;

FTI=F\_TransmissionsIDU\*100;

FTO=F\_TransmitOther\*100;

FE=Female\*100;

FEHC=Female\_HIVCases\*100;

FENH=Female\_New\_HIV\*100;

FPU=Female\_PREP\_Use\_P\*100;

FLU=Flu\_Vaccinated\*100;

FOI=Food\_Insecurity\*100;

FTHC=FTransmitHetero\_Contact\*100;

GOV=Gov\_Workers\*100;

G65=Greaterthan65\*100;

G100K=Grant\_Expenditure\_Center/100000;

HAST=HAStroke\_Treatment\_Center\*100;

HIS=Hispanic\*100;

HISH=Hispanic\_HIV\*100;

HISLC=Hispanic\_Latino\_Center\*100;

HIVC=HIV\_Center\*100;

HSCC=HIV\_Screening\_Center\*100;

HIVLCC=HIVLinkCare\_Center\*100;

HTEP=HIVTestEv\_Percent\*100;

HITYP=HIVTestyr\_Precent\*100;

HOM=Homeownership\*100;

HOP=Housing\_Problems\*100;

HSC=HS\_Completion\*100;

HSG=HS\_Grad\*100;

HYC=Hypertension\_Center\*100;

K12=K12\_Enrol\*100;

L18=Lessthan18\*100;

L5=Lessthan5\*100;

LCP=LinkCare\_percent\*100;

LC=Long\_Commute\*100;

LBW=Low\_Birth\_Weight\*100;

LBWC=Low\_Birth\_Weight\_Center\*100;

MHIVC=Male\_HIV\_Cases\*100;

MNH=Male\_New\_HIV\*100;

MPUP=Male\_PrEP\_Use\_Percent\*100;

MCC=Medicaid\_CHIP\_Center\*100;

MCE=Medical\_Center\*100;

MEC=Medicare\_Center\*100;

MHC=Mental\_Health\_Center\*100;

MIC=Minority\_Center\*100;

NAC=Native\_Center\*100;

NAHC=NativeHawaiianOther\_Center\*100;

NCDC=NC\_Diabetes\_Center\*100;

NM=Never\_Married\*100;

NBH=New\_Black\_HIV\*100;

NHH=New\_Hispanic\_HIV\*100;

NWH=New\_White\_HIV\*100;

NHCO=No\_Health\_Coverage\*100;

NHCP=NoHCP\_Percent\*100;

NH12P=NoHCP12\_Percent\*100;

OBE=Obesity\*100;

OAC=Older\_Adults\_Center\*100;

OIC=Other\_Insurance\_Center\*100;

OLC=Other\_Language\_Center\*100;

O1865=Over\_18\_Under\_65\*100;

PH=Pacific\_Hawaiian\*100;

PPFH=Percent\_Poor\_Fair\_Health\*100;

PPM=Percent\_Poor\_Mental14\*100;

PPP14=Percent\_Poor\_physical14\*100;

PP=Percent\_Poverty\*100;

PUH=Percent\_Unstable\_Housing\*100;

PI=Physical\_Inactivity\*100;

P25=Pop\_25\_noHSDiploma\*100;

PU34=PREP\_USE\_25\_34\*100;

PU24=PREPUSE\_13\_24\*100;

PU44=PREPUSE\_35\_44\*100;

PU54=PREPUSE\_45\_54\*100;

PU55=PREPUSE\_55\*100;

RC=Receipt\_Care\*100;

RP=Rural\_pop\*100;

RPU100KP=Rate\_PrEP\_Use\_100K/1000;

RCHP=Rate\_HIV\_Cases/1000;

RCHP12=HIV\_Rate/1000;

SHC=Severe\_Housing\_Cost\*100;

SMO=Smoker\*100;

STCC=Statin\_Therapy\_CVD\_Center\*100;

SAC=Substance\_Abuse\_Center\*100;

TC100K=Total\_Cost\_Center/100000;

TSC=TobaccoScreen\_Center\*100;

TMIDU=TransmissionMaleIDU\*100;

TMM=TransmistMMSexualContact\*100;

TMH=TransmitMHetero\_Contact\*100;

TMS=TransmitMM\_SexualIDU\*100;

TMO=TransmitMOther\*100;

TPR=Two\_Plus\_races\*100;

U65NI=Under65\_NoIns\*100;

UNE=Unemployment\*100;

UA=Uninsured\_Adults\*100;

UC=Uninsured\_Children\*100;

UCC=Uninsured\_Clinic\_Center\*100;

UGIS=Uninsured\_County\_GIS\*100;

UCH=Uninsured\_CountyHR\*100;

VH=Vacant\_Housing\*100;

VS=Virally\_Suppressed\*100;

VC=Vision\_Center\*100;

WHI=White\*100;

WHH=White\_HIV\*100;

WHO=White\_Only\*100;

DRAT=Dentist\_Ratio\*1000;

MHRAT=Mental\_Health\_HCP\_Ratio\*1000;

PCPRAT=PCP\_Ratio\*1000;

ORAT=Other\_PCPs\*1000;

RUN;

\* ;

\* The numargs macro was developed by Carrie Wager,Programmer,Channing Laboratory 1990;

\* Modified AMcD 1993, e hertzmark 1994 and L Chen 1996 ;

\* ;

%macro numargs(arg, delimit);

 %if %quote(&arg)= %then %do;

 0

 %end;

 %else %do;

 %let n=1;

 %do %until (%qscan(%quote(&arg), %eval(&n), %str( ))=%str());

 %let n=%eval(&n+1);

 %end;

 %eval(&n-1)

 %end;

 %mend numargs;

/\*\*\*\*\*\*\*\*\*\*\*\*\* get footnotes from table 1 call\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

%macro makefn();

%global fnlst fncnt;

%let fncnt = 0;

\*\*\*scan variable list to fncnt # in list;

%do %while(%qscan(&fn, &fncnt+1,%str(@)) ne %str());

 %let fncnt = %eval(&fncnt+1);

%end;

%do i = 1 %to &fncnt;

 %global fn&i fnvar&i;

 %let var&i = %Scan(&fn,&i,"@");

 %let fn&i =;

 %let fnvar&i =;

 data null;

 fnstr = "&&var&i";

 fn = prxchange("s/^\S+//", 1, fnstr);

 fnvar = prxchange("s/ \S+//", -1, fnstr);

 call symput("fn&i", fn);

 call symput("fnvar&i", fnvar);

 run;

%end;

%let fnlst=;

%do z = 1 %to &fncnt;

 %let fnlst = &fnlst &&fnvar&z;

%end;

%mend;

%macro t1rtf(

data =,

exposure=,

varlist=,

noadj =,

cat=,

mdn=,

rtftitle=,

landscape=,

fn=,

uselbl=,

file=,

dec=,

dec\_rnd=);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* end footnotes \*\*\*\*\*\*\*\*\*\*\*\*\*\*/

data table1dat;

set &data;

proc contents data = table1dat out = dsetcont noprint;

data dsetcont;

 set dsetcont;

 if missing(label) then label = name;

 lbltmp1 = prxchange('s/(BYTE\(\d{1,3}\))/%sysfunc($1)/I', -1, label);

 label = resolve(lbltmp1);

 drop lbltmp1;

run;

%let expfmt=;

%let explbl=;

/\* get labels and format name for levels of exposure variable \*/

proc sql noprint;

 select upcase(format) into :expfmt from dsetcont where upcase(name) = "%upcase(&exposure)";

 select label into :explbl from dsetcont where upcase(name) = "%upcase(&exposure)";

quit;

proc format;

 value levelfmt 0="&label0" 1="&label1" 2="&label2" 3="&label3" 4="&label4" 5="&label5"

 6="&label6" 7="&label7" 8="&label8" 9="&label9" 10="&label10" 11="&label11" 12="&label12"

 13="&label13" 14="&label14" 15="&label15";

/\* if missing format or request uselbl option, then use levelfmt option or defaults for

exposure level lables \*/

%if &expfmt eq %str() or %upcase("&uselbl")="T" %then %do; %let expfmt = LEVELFMT; %end;

/\* get formats using format name for exposure variable, to be used as labels \*/

proc format library = &fmtlib..&fmtcat cntlout=frmts;

data frmts;

set frmts;

if upcase(fmtname) = "%upcase(&expfmt)";

if type = "N" then do;

 startN = int(start); /\* start \* 1.0;\*/

 endN = int(end); /\*end \* 1.0;\*/

 fmttmp1 = prxchange('s/(BYTE\(\d{1,3}\))/%sysfunc($1)/I', -1, label);

 label = resolve(fmttmp1);

 drop fmttmp1;

end;

run;

/\* remove existing formats from exposure \*/

data table1dat;

 set table1dat;

 format &exposure;

run;

%let cntvar = 0;

/\*scan variable list to cntvar # in list - note: could change to numargs for consitiency left for now \*/

%do %while(%qscan(&varlist, &cntvar+1,%str( )) ne %str());

 %let cntvar = %eval(&cntvar+1);

%end;

/\* for footnote superscripts \*/

%let super = a b c d e f g h i j k l m n o p q r s t u v w x y z;

/\*\*\* M. Pazaris NOTE moved out of "v" loop \*\*\*/

%let count=0;

%let countobs=0;

%let expval =;

%let minexp=;

%let maxexp=;

proc sql noprint;

 select count(unique(&exposure)) into :count from table1dat where &exposure >= 0;

 select min(&exposure) into :minexp from table1dat where &exposure >= 0;

 select max(&exposure) into :maxexp from table1dat where &exposure >= 0;

 select unique(&exposure) into :expval separated by ' ' from table1dat where &exposure >= 0;

 select count(\*) into :countobs from table1dat where &exposure >= 0;

quit;

%let count = %sysfunc(compress(&count));

%let minexp = %sysfunc(compress(&minexp));

%let maxexp = %sysfunc(compress(&maxexp));

 %if "%upcase(&multn)" = "T" %then %do i = &minexp %to &maxexp;

 %let exp\_n&i =;

 proc sql;

 select %bquote(count(unique(&id))) into :exp\_n&i from table1dat where &exposure=&i;

 quit;

 %end;

/\*\*\* end M. Pazaris NOTE moved out of "v" loop \*\*\*/

data tab;

set tab;

%if &count > 1 %then %do; \_nall\_ = sum(of n&minexp-n&maxexp); %end;

%if &count = 1 %then %do; \_nall\_ = n&minexp; %end;

run;

%do v = 1 %to &cntvar;

%let var = %Scan(&varlist,&v," ");

%let varpol = %Scan(&varlist,%eval(&v+1)," ");

%let catv=;

%let mdnv=;

%let nadj=;

%let fn=;

%let mis=;

 /\*\*\*\* Calculate missing percent \*\*\*\*\*\*\*/

 %if "%upcase(&miscol)" = "T" %then %do;

          proc sql noprint;

 select (1-(\_nall\_/&countobs))\*100 into :mis from tab having upcase(varname) = "%upcase(&var)";

 quit;

 /\*\*\* Missing for polytomous categorical variabels \*\*\*\*/

 %if "&polycat" ne "" %then %do p = 1 %to &polyord;

 %if %upcase(&var) = %upcase(\_tmp&p) %then %do;

 proc sql noprint;

         select (1-(\_nall\_/&countobs))\*100 into :mis from tab having upcase(varname) = "%upcase(&varpol)";

 quit;

 %end;

 %end;

 %end;

%do c = 1 %to &cntvar;

%if %upcase(&var) eq %upcase(%Scan(&cat, &c," ")) %then %let catv = %Scan(&cat, &c," ");

%if %upcase(&var) eq %upcase(%Scan(&noadj, &c," ")) %then %let nadj = %Scan(&noadj, &c," ");

%if %upcase(&var) eq %upcase(%Scan(&fnlst, &c," ")) %then %let fn = &fn %Scan(&super, &c," ");

%if %upcase(&var) eq %upcase(%Scan(&mdn, &c," ")) %then %let mdnv = %Scan(&mdn, &c," ");

%end;

%let vn=0;

%let numpoly = %numargs(&poly);

\*\*\* iterate through each exposure value in list;

%do i = &minexp %to &maxexp; /\*1 %to &count;\*/

%let vn=%eval(&vn+1);

%let mrow&i=;

%let srow&i=;

%let arow&i=;

%let brow&i=;

%let crow&i=;

%let nrow&i=;

%let n&i=;

%let val = %Scan(&expval, &vn," ");

%let var\_lbl=;

%let lbl&i=;

%let lengthfmt =;

proc sql noprint;

%if %upcase(&var) ne %upcase(&catv) and %upcase(&var) ne %upcase(&nadj)

%then %do;

 select mn&i into :mrow&i from tab where upcase(varname)="%upcase(&var)";

 select std&i into :srow&i from tab where upcase(varname)="%upcase(&var)";

 select mdn&i into :arow&i from tab where upcase(varname)="%upcase(&var)";

 select p25&i into :brow&i from tab where upcase(varname)="%upcase(&var)";

 select p75&i into :crow&i from tab where upcase(varname)="%upcase(&var)";

 select label into :var\_lbl from dsetcont where upcase(name) = %upcase("&var");

%end;

%if %upcase(&var) ne %upcase(&catv) and %upcase(&var) eq %upcase(&nadj) %then %do;

 select mean(&var) into :mrow&i from table1dat where &exposure=&i;

 select std(&var) into :srow&i from table1dat where &exposure=&i;

 select label into :var\_lbl from dsetcont where upcase(name) = %upcase("&var");

%end;

%if %upcase(&var) eq %upcase(&catv) and %upcase(&var) ne %upcase(&nadj) %then %do;

 select n&i into :nrow&i from tab where upcase(varname)="%upcase(&var)";

 select mn&i\*100 into :mrow&i from tab where upcase(varname)="%upcase(&var)";

 select label /\*cats(label,', %')\*/ into :var\_lbl from dsetcont where upcase(name) = "%upcase(&var)";

%end;

%if %upcase(&var) eq %upcase(&catv) and %upcase(&var) eq %upcase(&nadj) %then %do;

 select n(&var) into :nrow&i from table1dat where &exposure=&i;

 select mean(&var)\*100 into :mrow&i from table1dat where &exposure=&i;

 select label /\*cats(label,', %')\*/ into :var\_lbl from dsetcont where upcase(name) = "%upcase(&var)";

%end;

select %bquote(count(\*)) FORMAT=40.0 into :n&i from table1dat where &exposure = &i;

select %bquote(label) into :lbl&i from frmts where upcase(fmtname) = "%upcase(&expfmt)" and startN <= &val <=endN;

quit;

%end;

%if &count > 5 or %eval(&count+&dec > 5) %then %do; %let landscape = T; %end;

%let nadjfn=;

%if "%upcase(&fn\_noadj)" ne "" %then %do i = 1 %to %numargs(&fn\_noadj);

 %if "%upcase(&var)" eq "%upcase(%scan(&fn\_noadj, &i, %str( )))" %then %do; %let nadjfn = %str(^{super \*});

 %end;

%end;

%\*put var = nadj &var = &&nadj;

data allrows&v;

%do j = &minexp %to &maxexp; /\* 1 %to &count;\*/

 length value $ 256;

 %if "%upcase(&var)" ne "%upcase(&catv)" %then %do;

 value = cats("&var\_lbl", "&nadjfn", "^{super &fn}"); %end;

 %if "%upcase(&var)" eq "%upcase(&catv)" %then %do;

 %if "%upcase(&pctn)" eq "N" %then %do;

 value = cats("&var\_lbl", "&nadjfn", "^{super &fn}",', n'); %end;

 %if "%upcase(&pctn)" eq "PCTN" %then %do;

 value = cats("&var\_lbl", "&nadjfn", "^{super &fn}",', %(n)'); %end;

 %if "%upcase(&pctn)" eq "PCT" %then %do;

 value = cats("&var\_lbl", "&nadjfn", "^{super &fn}",', %'); %end;

 %end;

 tmp\_n1 = symget("mrow&j");

 tmp\_n2 = symget("srow&j");

 tmp\_n3 = symget("arow&j");

 tmp\_n4 = symget("brow&j");

 tmp\_n5 = symget("crow&j");

 tmp\_n6 = symget("nrow&j");

if &dec < 1 then do;

 if abs(tmp\_n1) >= 1000 then do;

tmp\_n1b = round(tmp\_n1);

tmp\_n2b = round(tmp\_n2);

 tmp\_n3b = round(tmp\_n3);

 tmp\_n4b = round(tmp\_n4);

 tmp\_n5b = round(tmp\_n5);

 tmp\_n1bcat = tmp\_n1b;

tmp\_c1= put(tmp\_n1b, 30.0 -C);

tmp\_c2= put(tmp\_n2b, 30.0 -C);

tmp\_c3= put(tmp\_n3b, 30.0 -C);

 tmp\_c4= put(tmp\_n4b, 30.0 -C);

 tmp\_c5= put(tmp\_n5b, 30.0 -C);

 end;

 else if abs(tmp\_n1) >= 100 then do;

tmp\_n1b = round(tmp\_n1,.1);

tmp\_n2b = round(tmp\_n2,.1);

 tmp\_n3b = round(tmp\_n3,.1);

 tmp\_n4b = round(tmp\_n4,.1);

 tmp\_n5b = round(tmp\_n5,.1);

 tmp\_n1bcat = round(tmp\_n1);

tmp\_c1= put(tmp\_n1b, 30.1 -C);

tmp\_c2= put(tmp\_n2b, 30.1 -C);

 tmp\_c3= put(tmp\_n3b, 30.1 -C);

 tmp\_c4= put(tmp\_n4b, 30.1 -C);

 tmp\_c5= put(tmp\_n5b, 30.1 -C);

 end;

 else if abs(tmp\_n1) >= 10 then do;

tmp\_n1b = round(tmp\_n1,.01);

tmp\_n2b = round(tmp\_n2,.01);

 tmp\_n3b = round(tmp\_n3,.01);

 tmp\_n4b = round(tmp\_n4,.01);

 tmp\_n5b = round(tmp\_n5,.01);

 tmp\_n1bcat = round(tmp\_n1);

tmp\_c1= put(tmp\_n1b, 30.2 -C);

tmp\_c2= put(tmp\_n2b, 30.2 -C);

 tmp\_c3= put(tmp\_n3b, 30.2 -C);

 tmp\_c4= put(tmp\_n4b, 30.2 -C);

 tmp\_c5= put(tmp\_n5b, 30.2 -C);

 end;

 else if abs(tmp\_n1) >= 1 then do;

tmp\_n1b = round(tmp\_n1,.001);

tmp\_n2b = round(tmp\_n2,.001);

 tmp\_n3b = round(tmp\_n3,.001);

 tmp\_n4b = round(tmp\_n4,.001);

 tmp\_n5b = round(tmp\_n5,.001);

 tmp\_n1bcat = round(tmp\_n1);

tmp\_c1= put(tmp\_n1b, 30.3 -C);

tmp\_c2= put(tmp\_n2b, 30.3 -C);

 tmp\_c3= put(tmp\_n3b, 30.3 -C);

tmp\_c4= put(tmp\_n4b, 30.3 -C);

 tmp\_c5= put(tmp\_n5b, 30.3 -C);

 end;

 else if abs(tmp\_n1) >= 0 then do;

tmp\_n1b = round(tmp\_n1,.0001);

tmp\_n2b = round(tmp\_n2,.0001);

 tmp\_n3b = round(tmp\_n3,.0001);

 tmp\_n4b = round(tmp\_n4,.0001);

 tmp\_n5b = round(tmp\_n5,.0001);

 tmp\_n1bcat = round(tmp\_n1);

tmp\_c1= put(tmp\_n1b, 30.4 -C);

tmp\_c2= put(tmp\_n2b, 30.4 -C);

 tmp\_c3= put(tmp\_n3b, 30.4 -C);

 tmp\_c4= put(tmp\_n4b, 30.4 -C);

 tmp\_c5= put(tmp\_n5b, 30.4 -C);

 end;

 end;

if &dec >= 1 then do;

 tmp\_n1b = round(tmp\_n1, &dec\_rnd);

 tmp\_n1bcat = tmp\_n1b;

 tmp\_n2b = round(tmp\_n2, &dec\_rnd);

 tmp\_n3b = round(tmp\_n3, &dec\_rnd);

 tmp\_n4b = round(tmp\_n4, &dec\_rnd);

 tmp\_n5b = round(tmp\_n5, &dec\_rnd);

 tmp\_c1= put(tmp\_n1b, 30.&dec -C);

 tmp\_c2= put(tmp\_n2b, 30.&dec -C);

 tmp\_c3= put(tmp\_n3b, 30.&dec -C);

 tmp\_c4= put(tmp\_n4b, 30.&dec -C);

 tmp\_c5= put(tmp\_n5b, 30.&dec -C);

end;

 tmp\_n1bcat2=put(tmp\_n1bcat, 30.&dec -C);

tmp\_n6cat =round(tmp\_n6\*tmp\_n1/100);

if %upcase("&catv") ne %upcase("&var") then do;

 if "%upcase(&sep)" eq "PM" then do;

 var&j = compbl(tmp\_c1) || BYTE(177) || compbl(tmp\_c2);

 end;

 if "%upcase(&sep)" eq "PAR" then do;

 var&j =TRIM(LEFT(tmp\_c1))||" ("||TRIM(LEFT(tmp\_c2))||")";

 end;

end;

if %upcase("&catv") eq %upcase("&var") then do;

 if "%upcase(&pctn)" eq "N" then do;

 var&j=tmp\_n6cat;

 end;

 if "%upcase(&pctn)" eq "PCTN" then do;

 /\*var&j = compress(tmp\_n1bcat2 || "(" || tmp\_n6cat || ")");\*/

 var&j =TRIM(LEFT(tmp\_n1bcat2))||" ("||TRIM(LEFT(tmp\_n6cat))||")";

 end;

 if "%upcase(&pctn)" eq "PCT" then do;

 var&j = put(tmp\_n1bcat, 30.&dec -C);

 end;

end;

if %upcase("&mdnv") eq %upcase("&var") then do;

 var&j = trim(left(tmp\_c3))||" ("||trim(left(tmp\_c4))||"," ||trim(left(tmp\_c5))||")";

end;

/\*\*\*\*\*\*\*\* Format missing values for polytomous variables \*\*\*\*\*\*/

%if "&polycat" ne "" %then %do p1 = 1 %to %numargs(&allpoly);

 %if %upcase(%scan(&allpoly,&p1,%str( ))) = %upcase(&var) %then %do;

 value = '- ' || value;

 %let mis=9999;

 %end;

%end;

%if "&polycat" ne "" %then %do p2 = 1 %to &polyord;

 %if %upcase(&var) = %upcase(\_tmp&p2) %then %do;

 var&j = '00'x;

 %end;

%end;

%if "&poly" ne "" %then %do p3 =1 %to %numargs(&polylist);

 %if %upcase(%scan(&polylist,&p3,%str( ))) = %upcase(&var) %then %do;

 %let mis=9999;

 %end;

%end;

/\*\*\*\*\*\*\*\* End format missing values for polytomous varuiables \*\*\*\*\*\*/

if tmp\_c1 = . and tmp\_c3 = . then var&j = '00'x;

%if "%upcase(&multn)" ne "T" %then %do;

 %if "%upcase(&noexp)" ne "T" %then %do; label var&j = %sysfunc(cats("%unquote(&&lbl&j)", ' (n=', "%unquote(&&n&j)", ')' )); %end;

 %if "%upcase(&noexp)" eq "T" %then %do; label var&j = %sysfunc(cats(' (n=', "%unquote(&&n&j)", ')' )); %end;

%end;

%if "%upcase(&multn)" eq "T" %then %do;

 %if "%upcase(&noexp)" ne "T" %then %do; label var&j = %sysfunc(cats("%unquote(&&lbl&j)", ' (n\/N = ', "%unquote(&&exp\_n&j)",' \/ ', "%unquote(&&n&j)", ')' )); %end;

 %if "%upcase(&noexp)" eq "T" %then %do; label var&j = %sysfunc(cats(' (n\/N = ', "%unquote(&&exp\_n&j)",' \/ ', "%unquote(&&n&j)", ')' ));

%end;

%end;

 /\*\* set poly root = '' \*\*/

 %if "&poly" ne "" %then %do p=1 %to &numpoly;

 %if "%upcase(%scan(&poly,&p," "))" = "%upcase(&var)" %then %do; var&j = ''; %end;

 %end;

%end; /\*\* end J loop \*\*/

%if "%upcase(&miscol)" = "T" %then %do;

 if &mis ne 9999 then mis = put(&mis,4.1);

 else if "&mis" eq "9999" then mis = '00'x;

 label mis = 'Missing, %';

 /\*format mis 4.1;\*/

%end;

%\*put varz = "&var";

drop tmp\_n1 tmp\_n2 tmp\_c1 tmp\_c2 tmp\_n1b tmp\_n2b tmp\_n3 tmp\_n4 tmp\_n5 tmp\_c3 tmp\_c4 tmp\_c5 tmp\_n3b tmp\_n4b tmp\_n5b;

run;

proc append base = allrows data = allrows&v;

run;

%end; /\* end V loop \*/

data allrows;

set allrows;

label value = '00'x;

run;

title;

 libname tmplt './';

 ods path work.templat(update) sashelp.tmplmst(read);

proc template;

 define style Styles.Custom;

 parent = Styles.RTF;

replace Table from Output /

 frame = hsides /\* outside borders: void, box, above/below, vsides/hsides, lhs/rhs \*/

 rules = groups /\* internal borders: none, all, cols, rows, groups \*/

 cellpadding = 3pt /\* the space between table cell contents and the cell border \*/

 cellspacing = 0pt /\* the space between table cells, allows background to show \*/

 borderwidth = .75pt /\* the width of the borders and rules \*/;

replace color\_list /

 'link' = blue /\* links \*/

 'bgH'= white

 'fg' = black /\* text color \*/

 'bg' = white; /\* page background color \*/;

replace fonts /

 'TitleFont' = ("Times Roman",12pt) /\* Titles from TITLE statements \*/

 'TitleFont2' = ("Times Roman",12pt,Bold Italic) /\* Procedure titles ("The \_\_\_\_\_ Procedure")\*/

 'StrongFont' = ("Times Roman",10pt,Bold)

 'EmphasisFont' = ("Times Roman",10pt,Italic)

 'headingEmphasisFont' = ("Times Roman",11pt,Bold Italic)

 'headingFont' = ("Times Roman",12pt) /\* Table column and row headings \*/

 'docFont' = ("Times Roman",12pt) /\* Data in table cells \*/

 'footFont' = ("Times Roman",12pt) /\* Footnotes from FOOTNOTE statements \*/

 'FixedEmphasisFont' = ("Courier",9pt,Italic)

 'FixedStrongFont' = ("Courier",9pt,Bold)

 'FixedHeadingFont' = ("Courier",9pt,Bold)

 'BatchFixedFont' = ("Courier",6.7pt)

 'FixedFont' = ("Courier",9pt);

end;

run;

goptions reset=all;

options papersize=A4 %if %upcase(&landscape) = T %then %do; orientation=landscape %end;

%if %upcase(&landscape) ^= T %then %do; orientation=portrait %end;;

ods listing close;

ods rtf file="./&file..doc" style=Custom;

ods escapechar='^';

proc report data = allrows nowd style(header)={just=LEFT} /\*style(report)={outputwidth=6.5 in}\*/;

column ("{\b Table 1 \b0 %str(&rtftitle)}" value ("%str(&explbl)" %do z = &minexp %to &maxexp /\*1 %to &count\*/; var&z

%end;

%if "%upcase(&miscol)" = "T" %then %do;

mis

%end;));

define value / display ' ' /\*style(column)={cellwidth=2in}\*/;

%do z = &minexp %to &maxexp; /\*1 %to &count;\*/

/\*define var&z / noprint\*/;

define var&z / display;

%end;

run;

quit;

%if "%upcase(&sep)" = "PM" %then %do;

 %if "%upcase(&ageadj)" ne "F" %then %do; ODS rtf text = "^S={Font=('Times Roman',12pt)}Values are means %sysfunc(BYTE(177)) SD or

 percentages and are standardized to the age distribution of the study population."; %end;

 %if "%upcase(&ageadj)" eq "F" %then %do; ODS rtf text = "^S={Font=('Times Roman',12pt)}Values are means %sysfunc(BYTE(177)) SD

 or percentages."; %end;

%end;

%if "%upcase(&sep)" = "PAR" %then %do;

 %if "%upcase(&ageadj)" ne "F" %then %do; ODS rtf text = "^S={Font=('Times Roman',12pt)}Values are means(SD) or medians(Q25, Q75) for continuous variables; percentages or ns or both for categorical variables, and are

standardized to the age distribution of the study population."; %end;

 %if "%upcase(&ageadj)" eq "F" %then %do; ODS rtf text = "^S={Font=('Times Roman',12pt)}Values are means(SD) or medians(Q25, Q75) for continuous variables; percentages or ns or both for categorical variables."; %end;

%end;

%if "&poly" ne "" or "&polycat" ne "" %then %do;

 ODS rtf text = "^S={Font=('Times Roman',12pt)} Values of polytomous variables may not sum to 100% due to rounding"; %end;

%do z = 1 %to &fncnt;

ODS rtf text = "^{super %Scan(&super, &z," ")} &&fn&z";

%end;

%if "%upcase(&ageadj)" eq "T" and "%upcase(&fn\_noadj)" ne "" %then %do;

ODS rtf text = "^S={Font=('Times Roman',12pt)} ^{super \*} Value is not age adjusted"; %end;

quit;

ods rtf close;

goptions reset=all;

ods listing;

quit;

proc datasets nolist;

 delete allrows; run; quit;

%mend;

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

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

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Poly Means Macro \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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

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

%macro polymeans(polyvars=, data=);

%local polyvars fmtname vlbl fmtvals fmtlbls minfmt maxfmt adjpoly;

%let pcount = 0;

\*\*\*scan variable list to pcount # in list;

%do %while(%qscan(&polyvars, &pcount+1,%str( )) ne %str());

 %let pcount = %eval(&pcount+1);

%end;

%local polyorig;

%let polyorig = &polyvars;

%global polylist;

%let polylist = &polylist;

proc contents data = &data out = dsetcont noprint;

%do pol = 1 %to &pcount;

%let adjpoly=F;

%let polyvar = %Scan(&polyvars,&pol," ");

%if "&noadj" ne "" %then %do a = 1 %to %numargs(&noadj);

%if "%upcase(%scan(&noadj, &a, %str( )))" eq "%upcase(&polyvar)" %then %do;

 %let adjpoly = T;

 %end;

%end;

%let fmtname=;

%let vlbl=;

proc sql noprint;

select format into :fmtname from dsetcont where upcase(name) = "%upcase(&polyvar)";

select label into :vlbl from dsetcont where %upcase(name) = "%upcase(&polyvar)";

quit;

proc format cntlout=frmts;

%let fmtvals=;

%let fmtlbls=;

%let minfmt=;

%let maxfmt=;

proc sql noprint;

select start into :fmtvals from frmts where upcase(fmtname) = "%upcase(&fmtname)";

select label into :fmtlbls separated by "^^^" from frmts where upcase(fmtname) = "%upcase(&fmtname)";

select min(start) into :minfmt from frmts where upcase(fmtname) = "%upcase(&fmtname)";

select max(start) into :maxfmt from frmts where upcase(fmtname) = "%upcase(&fmtname)";

quit;

%let minfmt = %sysfunc(compbl(&minfmt));

%let maxfmt = %sysfunc(compbl(&maxfmt));

%let fmtlbls&pol=&fmtlbls;

%let plst =;

%if &pol=1 %then %do;

 %let polorig\_S = %sysfunc(compbl(&minfmt));

 %let polorig\_E = %sysfunc(compbl(&maxfmt));

%end;

%if &pol>1 %then %do;

 %let polorig\_S = %sysfunc(compbl(&polorig\_S &minfmt));

 %let polorig\_E = %sysfunc(compbl(&polorig\_E &maxfmt));

%end;

%do j = &minfmt %to &maxfmt;

 %let mgt0 = 0;

 %let polylist = &polylist \_tmp&pol.\_&j;

 proc sql noprint;

 select mean(&polyvar) into :mgt0 from &data having &polyvar = &j;

 quit;

 %if &mgt0 > 0 %then %do; %let plst = &plst \_tmp&pol.\_&j; %end;

%end;

%if &pol=1 %then %do;

 %let polyarray = %sysfunc(compbl(%str(array &polyvar.\_a {\*} \_tmp&pol.\_&minfmt - \_tmp&pol.\_&maxfmt;)));

 %let parrayname = %sysfunc(compbl(&polyvar.\_a{j}));

 %let parraynum = %sysfunc(compbl(%eval(&maxfmt - &minfmt + 1)));

%end;

%if &pol>1 %then %do;

 %let polyarray = &polyarray %sysfunc(compbl(%str(array &polyvar.\_a {\*} \_tmp&pol.\_&minfmt - \_tmp&pol.\_&maxfmt;)));

 %let parrayname = %sysfunc(compbl(&parrayname &polyvar.\_a{j}));

 %let parraynum = %sysfunc(compbl(&parraynum %eval(&maxfmt - &minfmt + 1)));

%end;

%if &pol=1 %then %do; %let v1=%numargs(&varlist); %let tmpvarlist=; %let startvar=1; %end;

%if &pol > 1 %then %do; %let v1=%numargs(&varlist); %end;

%let stopvar=0;

%do vlst = &startvar %to &v1;

 %if &stopvar=0 and "%upcase(%scan(&varlist, &vlst, %str( )))" ne "%upcase(&polyvar)" %then %do;

 %let tmpvarlist = &tmpvarlist %scan(&varlist, &vlst, %str( ));

 %end;

 %if &stopvar=0 and "%upcase(%scan(&varlist, &vlst, %str( )))" eq "%upcase(&polyvar)" %then %do;

 %let tmpvarlist = &tmpvarlist %scan(&varlist, &vlst, %str( )) &plst;

 %let cat = &cat &plst;

 %if "%upcase(&adjpoly)" = "T" %then %do;

 %let noadj = &noadj &plst;

 %end;

 %if &pol < &pcount %then %do; %let stopvar=1; %let startvar=%eval(&vlst+1); %end;

 %end;

 %end;

%end;

/\*%put pazaris polyorig=&polyorig polyarray = &polyarray parrayname = &parrayname parraynum =&parraynum;

varlist=&varlist cat=&cat plst=&plst polorig\_S=&polorig\_S polorig\_E=&polorig\_E \*/

%let varlist = &tmpvarlist;

data &data;

set &data;

 array origvar {\*} &polyorig;

 &polyarray;

 %do i = 1 %to %numargs(&polyorig);

 numto = (scan("&parraynum",&i," "));

 do j = 1 to numto;

 if ^missing(origvar{&i}) then %scan("&parrayname", &i, " ") = 0;

 if origvar{&i}=j then %scan("&parrayname", &i, " ") = 1;

 end;

 %do k = 1 %to %scan("&parraynum",&i," ");

 label \_tmp&i.\_&k = "- %scan(&&fmtlbls&i, &k, %str(^^^))";

 %end;

 %end;

run;

%mend;

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

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* end poly means macro \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* start poly cat macro \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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

%macro polycatset();

%global allpoly polyord;

%let polyord=0;

%let polycnt=0;

%let allpoly=;

%let polycatcnt=0;

\*\*\*scan variable list to polycat # in list;

%do %while(%qscan(&polycat, &polycatcnt+1,%str(@)) ne %str());

 %let polycatcnt = %eval(&polycatcnt+1);

%end;

%let polystr=;

%do i = 1 %to &polycatcnt;

%let indexcat =;

%let adjrem =;

%let polystr=%Scan(&polycat,&i,"@");

%let polylbl&i=%Scan(&polystr,1,"$");

%let polycv=%Scan(&polystr,2,"$");

%let allpoly = &allpoly &polycv;

%let polyord = %scan(&polystr,1,%str( ));

%if &i eq 1 %then %do; %let polyord2 = &polyord; %end;

%if &i ne 1 %then %do; %let polyord2 = %eval(&polyord + &polycnt); %end;

%let polylbl&i =

%sysfunc(compbl(%substr(&&polylbl&i,%eval(%length(&polyord)+1),%eval(%length(&&polylbl&i)-%length(&polyord)))));

%let polycnt = %eval(%numargs(&polycv) + &polycnt);

%let varcnt = %numargs(&varlist);

%do z = 1 %to %numargs(&polycv);

%let tmpv = %Scan(&polycv,&z,%str( ));

 %macro skip;

 data &data;

 set &data;

 /\*label &tmpv = "- ";\*/

 run;

 %mend;

%end;

 /\* Create list of no-adjustment variables for not adjusted footnote \*/

 %let indexcat = %index(%upcase(&fn\_noadj), %upcase(&polycv));

 %if &indexcat > 0 %then %let adjrem = %substr(&fn\_noadj, &indexcat, %length(&polycv));

 %if &indexcat = 1 and (%eval(&indexcat+%length(&polycv)-1) = %length(&fn\_noadj)) %then %let fn\_noadj = \_tmp&i;

%let index1 = %eval(&indexcat+%length(&polycv);

%let index2 = %length(&fn\_noadj);

%\*put index 1 is &index1 and index2 is &index2;

 %if &indexcat = 1 and (%eval(&indexcat+%length(&polycv)-1) < %length(&fn\_noadj)) %then %let fn\_noadj =

\_tmp&i %substr(&fn\_noadj, %eval(&indexcat+%length(&polycv)-1));

 %if &indexcat > 1 %then %let fn\_noadj = \_tmp&i %substr(&fn\_noadj, 1, %eval(&indexcat-1)) %substr(&fn\_noadj, %eval(&indexcat+%length(&polycv)-1));

 %if %length(&fn\_noadj) > 0 %then %let fn\_noadj = %sysfunc(compbl(&fn\_noadj));

%let varlist2 =;

%do z2 = 1 %to &varcnt;

 %if &z2 ne &polyord2 %then %do;

 %let varlist2 = &varlist2 %scan(&varlist,&z2, %str( ));

 %end;

%if &z2 eq &polyord2 %then %do;

 %let varlist2 = &varlist2 \_tmp&i &polycv %scan(&varlist, &z2, %str( ));

 %end;

%end;

%if &polyord2 > &varcnt %then %do;

 %let varlist2 = &varlist2 \_tmp&i &polycv;

%end;

%put polyord2 = &polyord2 varcnt = &varcnt varlist = &varlist varlist2 = &varlist2;

%let varlist = &varlist2;

%let cat = &cat &polycv;

%end;

/\*\* M. Pazaris rearanged to avoide multiple datastep sets \*\*/

data &data;

set &data;

 %do i2 = 1 %to &polycatcnt;

 \_tmp&i2=.;

 label \_tmp&i2="&&polylbl&i2";

%end;

run;

%mend;

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

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* end poly cat macro \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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

/\*This program was designed to compute direct standardization of

 rates, means, or proportions. Specifically, Table 1 of many papers

 is a breakdown of cohort characteristics by exposure categories. In

 most instances, it is necessary to age-standardize the means or

 proportions of other potential confounders before displaying them

 by exposure category.

 changes from eric's program:

 requires exposure to be nonmissing

 allows more than 1 vbl in covar list

 computes std dev on weighted original data. this estimates

 the std the group would have if it had had the same age distribution

 as the standard.

 also computes std error of the standardized mean. this assumes that

 the weights relating to the means are constants, and that we do not

 take the between-age-group variance into account.

 \*/

%macro table1(data=,

 ageadj=T,

 exposure=,

 noexp=F,

 agegroup=,

 varlist=,

 header=,

 missing=.,

 covar=,

 printvar=mean std se ,

 extstand=,

 explab=,

 label0=level 0,

 label1=level 1, label2=level 2, label3=level 3,

 label4=level 4, label5=level 5,

 label6=level 6, label7=level 7, label8=level 8,

 label9=level 9, label10=level 10,

 label11=level 11, label12=level 12, label13=level 13,

 label14=level 14, label15=level 15,

 notes=nonotes,

 /\*\*\* rtf options \*\*\*/

 nortf=F,

 noadj=,

 cat=,

 rtftitle=,

 landscape=F,

 fn=,

 uselbl=F,

 file=,

 miscol=F,

 poly=,

 mdn=,

 polycat=,

 fmtlib=work,

 fmtcat=formats,

 multn=F,

 id=id,

 sep = par,

 pctn=pct,

 dec=0);

 options &notes;

/\*\*\*\* resort poly needs to match order of varlist \*\*\*\*/

%if &varlist ne and &poly ne %then %do;

%let varlist = %sysfunc(compbl(&varlist));

%let poly = %sysfunc(compbl(&poly));

%let newpoly =;

%let vpv=;

%let vpp=;

%do i = 1 %to %numargs(&varlist);

 %let vpv=%scan(&varlist, &i, %str( ));

 %do j = 1 %to %numargs(&poly);

 %let vpp=%scan(&poly, &j, %str( ));

 %if "%upcase(&vpv)" = "%upcase(&vpp)" %then %do;

 %let newpoly = &newpoly &vpv;

 %end;

 %end;

 %end;

%let poly = &newpoly;

%end;

/\*\*\*\* end resort poly needs to match order of varlist \*\*\*\*/

/\* table1 can't be age-adjusted if noexp=T\*/

%if %str(%upcase(&noexp)) eq %str(T) %then %let ageadj=F;

/\*\*\*\*\*\*\*\* TEST FOR MISSING VARIABLES IN VARLIST \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

proc contents data = &data out = \_tmp\_vcol\_ noprint;

run;

%let missingvar =;

%do i = 1 %to %numargs(&varlist);

 %let var2chk = %scan(&varlist, &i, %str( ));

 %let varnumchk =;

 proc sql noprint;

 select varnum into:varnumchk from \_tmp\_vcol\_

 having /\*upcase(memname)=upcase("&data") and\*/ upcase(name) = upcase("&var2chk");

 quit;

 %if "&varnumchk" eq "" %then %do; %let missingvar = &missingvar &var2chk; %end;

%\*put pazaris test varnumchk = &varnumchk;

%end;

/\*\*\*\*\*\*\*\* END TEST FOR MISSING VARIABLES IN VARLIST \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\* TEST FOR DUPLICATE VARIABLES IN VARLIST \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

%let dupvalue =;

%do k1 = 1 %to %eval(%numargs(&varlist)-1);

 %let firstvar = %scan(&varlist, &k1, %str( ));

 %do k2 = %eval(&k1+1) %to %numargs(&varlist);

 %let secondvar = %scan(&varlist, &k2, %str( ));

 %if %upcase(&firstvar) = %upcase(&secondvar) %then %do;

 %let dupvalue = &dupvalue &firstvar;

 %end;

 %end;

%end;

/\*\*\*\*\*\*\*\* END TEST FOR DUPLICATE VARIABLES IN VARLIST \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

proc format library= &fmtlib..&fmtcat;

value \_tbl1\_fmt\_creat\_val\_ 1='yes' 2='no';

proc format library = &fmtlib..&fmtcat cntlout=frmtschg;

data frmtschg;

length label $ 100;

set frmtschg;

run;

data frmtschg;

length label $ 100;

set frmtschg;

 label = prxchange("s/,/BYTE(130)/", -1, label);

run;

proc format library = &fmtlib..&fmtcat cntlin=frmtschg;

run;

%let errcall = 0;

%if %str(&data) = %str() %then %do;

%put ERROR: You did not provide a data-set name.;

%goto exit;

%end;

%if %str(%upcase(&noexp)) ne %str(T) and %str(&exposure) = %str() %then %do;

%put ERROR: You did not provide an exposure variable.;

%put ERROR: If you do not want an exposure, please set noexp=T.;

%goto exit;

%end;

%if %str(%upcase(&noexp)) ne %str(T) and %numargs(&exposure) > 1 %then %do;

%put ERROR: You can only have one exposure variable.;

%goto exit;

%end;

%if %str(&agegroup) = %str() and ((%str(%upcase(&ageadj)) ne %str(F)) and (%str(%upcase(&noexp)) ne %str(T))) %then %do;

%put ERROR: You did not provide a varaiable for age-adjustment.;

%goto exit;

%end;

%if %str(&file) = %str() and %str(%upcase(&nortf)) ne %str(T) %then %do;

%put ERROR: You did not provide a file name for the MS Word table.;

%goto exit;

%end;

%if %str(&varlist) = %str() %then %do;

%put ERROR: You did not provide a list of variables.;

%goto exit;

%end;

%if %numargs(&varlist) = 1 %then %do;

%put ERROR: You need to provide more than one variable in varlist.;

%goto exit;

%end;

%if "&missingvar" ne "" %then %do;

%put ERROR: You have included nonexistent variable(s) &missingvar in the table1 call.;

%goto exit;

%end;

%if "&dupvalue" ne "" %then %do;

%put ERROR: You have included duplicate variable(s) &dupvalue in the table1 call.;

%goto exit;

%end;

/\*\*\*\*\*\*\* test for 0,1 coding for categorical variables \*\*\*\*\*\*\*\*\*\*\*\*\*/

%do c = 1 %to %numargs(&cat);

%let cvariable = %scan(&cat, &c, %str( ));

%let minc=;

%let maxc=;

 proc sql noprint;

 select min(&cvariable) into:minc from &data;

 select max(&cvariable) into:maxc from &data;

 quit;

%if &minc ne 0 or &maxc ne 1 %then %do;

 %put ERROR: categorical variables must be coded 0 1.;

 %put ERROR: variable &cvariable is coded as &minc &maxc.;

 %goto exit;

 %end;

%end;

/\*\*\*\*\*\*\* end test for 0,1 coding for categorical variables \*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\* test for noadj for age adjusted distributions \*\*\*\*\*\*\*\*\*\*\*\*\*/

%let noadjmdn =;

%do k1 = 1 %to %numargs(&mdn);

 %let mdnvar = %scan(&mdn, &k1, %str( ));

 %do k2 = 1 %to %numargs(&noadj);

 %let noadjvar = %scan(&noadj, &k2, %str( ));

 %if %upcase(&mdnvar) = %upcase(&noadjvar) %then %do;

 %let noadjmdn = &noadjmdn &noadjvar;

 %end;

 %end;

%end;

%if "&noadjmdn" ne "" %then %do;

%put ERROR: Remove &noadjmdn from mdn= Do a separate table1 macro call for &noadjmdn without age adjustment.;

%goto exit;

%end;

/\*\*\*\*\*\*\* end: test for noadj for age adjusted distributions \*\*\*\*\*\*\*\*\*\*\*\*\*/

%let dec\_rnd = %sysevalf(1 \* 10\*\*-&dec);

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

data tbl1adj;

set &data;

%if "%upcase(&ageadj)" = "F" %then %do; t1agegrp=1; %end; /\* if table is not to be age

adjusted, set age adjustment variable to 1 \*/

%if "%upcase(&noexp)" = "T" %then %do; t1defaultexp=1; label t1defaultexp='00'x; %end; /\* create dummy default exposure if no exposure

requested \*/

run;

%if "%upcase(&ageadj)" = "F" %then %do; %let agegroup=t1agegrp; %end;

%let data = tbl1adj;

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

%if %str(&varlist) ne %str() %then %let varlist=%sysfunc(compbl(&varlist));

%if %str(&covar) ne %str() %then %let covar=%sysfunc(compbl(&covar));

%if %str(&noadj) ne %str() %then %let noadj=%sysfunc(compbl(&noadj));

%if %str(&cat) ne %str() %then %let cat=%sysfunc(compbl(&cat));

%if %str(&poly) ne %str() %then %let poly=%sysfunc(compbl(&poly));

%if %str(&polycat) ne %str() %then %let polycat=%sysfunc(compbl(&polycat));

%if %str(&mdn) ne %str() %then %let mdn=%sysfunc(compbl(&mdn));

%macro changelist(list=, listm=, datarename=);

%let list2 =;

%do i = 1 %to %numargs(&list);

 %if "%upcase(%scan(&list, &i, %str( )))" = "CASE" %then %do;

 %let list2 = &list2 \_case;

 %let tmplbl =;

 %if "%upcase(&datarename)" = "T" %then %do;

 data &data;

 set &data;

 call symput('tmplbl', vlabel(%scan(&list, &i, %str( ))));

 run;

 data &data;

 set &data;

 \_case = case;

 label \_case = " &tmplbl";

 run;

 %end;

 %end;

 %if "%upcase(%scan(&list, &i, %str( )))" ne "CASE" %then %do;

 %let list2 = &list2 %scan(&list, &i, %str( ));

 %end;

 %end;

%let &listm = &list2;

%mend changelist;

%changelist(list=&agegroup, listm=agegroup, datarename=T);

%changelist(list=&exposure, listm=exposure, datarename=T);

%changelist(list=&varlist, listm=varlist, datarename=T);

%changelist(list=&cat, listm=cat, datarename=F);

%changelist(list=&poly, listm=poly, datarename=F);

%changelist(list=&polycat, listm=polycat, datarename=T);

%changelist(list=&mdn, listm=mdn, datarename=F);

/\*\*\*\*\*\*\* check to make sure poly variable is in varlist \*\*\*\*\*/

%if "&poly" ne "" %then %do pv = 1 %to %numargs(&poly);

%let polyvarerr=0;

 %let plyvar = %scan(&poly, &pv, %str( ));

 %do pv2 = 1 %to %numargs(&varlist);

 %if %upcase(&plyvar) = %upcase(%scan(&varlist, &pv2, %str( ))) %then %let polyvarerr=1;

 %end;

 %if &polyvarerr=0 %then %do;

 %put ERROR: all poly variables must be in varlist in order to determine table1 variable order.;

 %put ERROR: poly variable &plyvar was not found in varlist.;

 \*%goto exit;

 %end;

%end;

%if &errcall = 1 %then %goto exit;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Check poly formats \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 proc contents data = &data out = dset4miss noprint;

 proc format library = &fmtlib..&fmtcat cntlout=fm4miss;

 data fm4miss;

 set fm4miss;

 startN = int(start); /\* start \* 1.0;\*/

 endN = int(end); /\*end \* 1.0;\*/

 run;

%if "&poly" ne "" %then %do p = 1 %to %numargs(&poly);

%let ptstvar = %scan(&poly, &p, %str( ));

%let minptv =;

%let maxptv =;

proc sql noprint;

select min(&ptstvar) into :minptv from &data where ^missing(&ptstvar);

select max(&ptstvar) into :maxptv from &data where ^missing(&ptstvar);

quit;

%\*put mpazaris pts test ptstvar = &ptstvar minptv = &minptv maxptv = &maxptv;

 %\*if &minptv ^= 1 %then %do;

 %\*put ERROR poly variables lowest value must be 1;

 %\*put the lowest value for &ptstvar is &minptv;

 %\*let errcall = 1;

 %\*end;

 %let fmt4miss=;

 proc sql noprint;

 select upcase(format) into :fmt4miss from dset4miss where upcase(name) = "%upcase(&ptstvar)";

 quit;

 /\*%if &minptv = 1 %then\*/ %do mm = 1 %to &maxptv;

 %let missfmt = ;

 proc sql noprint;

 select fmtname into :missfmt from fm4miss where upcase(fmtname) = "%upcase(&fmt4miss)" and startN = &mm;

 quit;

 %if "&missfmt" = "" %then %do;

 %put ERROR: poly variables must be formated consecutively starting at 1.;

 %put ERROR: there is no format for level &mm for variable &ptstvar.;

 \*%goto exit;

 %end;

 %end;

%end;

%if &errcall = 1 %then %goto exit;

/\* end check poly formats \*/

%let fn\_noadj = &noadj;

%if %length(&rtftitle) > 0 %then %let rtftitle = %sysfunc(compbl(&rtftitle));

%if %length(&fn) > 0 %then %let fn = %sysfunc(compbl(&fn));

%makefn();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* create polytomous variables \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* create polytomous variables automatically when supplied with a ordinal variable \*/

%if "&poly" ne "" %then %do; %polymeans(polyvars=&poly, data=&data); %end;

/\* use list of pre-defined polytomous variables \*/

%if "&polycat" ne "" %then %do; %polycatset(); %end;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* end create polytomous variables \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

%if "%upcase(&noexp)" = "T" %then %do; %let exposure=t1defaultexp; %let ageadj=F; %end;

 options &notes;

%local nvar i \_minexp\_ \_maxexp\_ \_nlev\_ ;

%let nvar=%numargs(&varlist);

/\* this is the analysis data set. no COVAR missing, no exposure missing \*/

data \_tmp\_;

 set &data;

 where &exposure ne &missing;

 %if "&covar" ne "" %then %do;

 if nmiss(of &covar) eq 0;

 %end;

 run;

/\* this gets the levels of exposure, assuming they go by 1 from minexp to maxexp \*/

proc means noprint data=\_tmp\_; var &exposure;

 output out=\_minmax\_ min=minexp max=maxexp;

 run;

data \_minmax\_; set \_minmax\_;

 call symput('\_minexp\_', trim(left(minexp))); call symput('\_maxexp\_', trim(left(maxexp)));

 nlev=1+maxexp-minexp;

 call symput('\_nlev\_',trim(left(nlev)));

 run;

/\* if not using an external standard age distribution \*/

/\* then mainfreq is the age distribution of the whole study population \*/

%if %length(&extstand) eq 0 %then %do;

proc freq data=\_tmp\_;

 tables &agegroup/noprint out=mainfreq;

 run;

data mainfreq; set mainfreq; pctov=percent;

 keep pctov &agegroup;

 run;

/\* remember that pctov is a percent, not a fraction \*/

%end;

/\* using an external standard age distribution.

data set must have variables &agegroup and pctov (as a percent) \*/

%if %length(&extstand) ne 0 %then %do;

 data mainfreq; set &extstand; keep pctov &agegroup; run;

%end;

proc sort data=\_tmp\_; by &exposure; run;

/\* this is to get the actual fractions of each exposure level in each age group \*/

proc freq data=\_tmp\_; tables &agegroup / out=\_tmp1\_ noprint; by &exposure; run;

data \_tmp1\_; set \_tmp1\_; pctexp=percent; keep pctexp &agegroup &exposure; run;

proc sort data=\_tmp1\_; by &agegroup; run;

proc sort data=\_tmp\_; by &agegroup &exposure; run;

proc sort data=mainfreq; by &agegroup; run;

data \_tmp1\_; merge mainfreq \_tmp1\_; by &agegroup;

 if pctexp gt 0 then wgt=pctov/pctexp;

 else do;

 put "WARNING from macro: No observations in agegroup &agegroup , exposure level &exposure .";

 file print;

 put "WARNING from macro: No observations in agegroup &agegroup , exposure level &exposure .";

 end;

 run;

proc sort data=\_tmp1\_; by &agegroup &exposure; run;

data \_tmp\_; merge \_tmp1\_ \_tmp\_; by &agegroup &exposure; run;

/\* this gets standardized means and standard deviation assuming the

group had the age distribution of the standard population \*/

proc sort data=\_tmp\_; by &exposure &agegroup; run;

proc means noprint data=\_tmp\_ n mean std median p25 p75;

 var &varlist; weight wgt; by &exposure;

 output out=\_stdm\_

 %if &nvar eq 1 %then %do;

 mean=mean std=std n=n median=median p25=p25 p75=p75;

 %end;

 %else %do;

 mean=mean1-mean&nvar std=std1-std&nvar n=n1-n&nvar

 median=median1-median&nvar p25=p251-p25&nvar p75=p751-p75&nvar ;

 %end;

 run;

/\* now get std errors \*/

proc means noprint data=\_tmp\_;

var &varlist;

output out=descrips

 %if &nvar eq 1 %then %do; std=std; %end;

 %else %do; std=std1-std&nvar; %end;

by &exposure &agegroup;

run;

proc sort data=descrips; by &agegroup; run;

data descrips; merge descrips mainfreq; by &agegroup;

 wtse=pctov\*pctov/10000;

%if &nvar eq 1 %then %do; s1=std1\*std1\*wtse; %end;

%else %do;

 array stds{\*} std1-std&nvar;

 array summands{\*} s1-s&nvar;

 do i=1 to dim(stds); summands{i}=wtse\*stds{i}\*stds{i}; end;

%end;

run;

proc sort; by &exposure; run;

proc means noprint data=descrips;

 %if &nvar eq 1 %then %do; var s1;

 output out=stderrsq sum=v1;

 %end;

 %else %do; var s1 - s&nvar ;

 output out=stderrsq sum=v1 - v&nvar ;

 %end;

by &exposure;

run;

data stderrsq; set stderrsq;

%if &nvar eq 1 %then %do; if v1 gt 0 then se1=sqrt(v1); %end;

%else %do;

 array vs{\*} v1-v&nvar;

 array ses{\*} se1-se&nvar;

 %do i=1 %to &nvar; if vs{&i} gt 0 then ses{&i}=sqrt(vs{&i}); %end;

%end;

keep &exposure

 %if &nvar eq 1 %then %do; se1 ; %end;

 %else %do; se1 - se&nvar ; %end;

run;

 /\* doing rounding and putting out one obs per variable \*/

data \_stdm\_; merge \_stdm\_ stderrsq; by &exposure;

length varname $32 /\*$20\*/;

%if &nvar eq 1 %then %do;

 varname=trim("&varlist");

%end;

%else %do;

 array ns{\*} n1-n&nvar;

 array mns{\*} mean1-mean&nvar;

 array stds{\*} std1-std&nvar;

 array ses{\*} se1-se&nvar;

 array mdns{\*} median1-median&nvar;

 array p25s{\*} p251-p25&nvar;

 array p75s{\*} p751-p75&nvar;

 %do i=1 %to &nvar;

varname=trim(left("%scan(&varlist, &i, %str( ))"));

 ord=.; ord=trim(left(&i));

 n=trim(left(ns{&i})); mean=trim(left(mns{&i})); std=trim(left(stds{&i})); se=trim(left(ses{&i}));

median=trim(left(mdns{&i})); p25=trim(left(p25s{&i})); p75=trim(left(p75s{&i}));

 output;

 %end;

%end;

keep &exposure varname n mean std se median p25 p75 ord ;

 run;

proc sort data=\_stdm\_; by ord varname &exposure;

data \_stdm\_; set \_stdm\_;

 minexp=.; maxexp=.; minexp=&\_minexp\_; maxexp=&\_maxexp\_;

 array ns{&\_minexp\_ : &\_maxexp\_} n&\_minexp\_ - n&\_maxexp\_;

 array mns{&\_minexp\_ : &\_maxexp\_} mn&\_minexp\_ - mn&\_maxexp\_;

 array stds{&\_minexp\_ : &\_maxexp\_} std&\_minexp\_ - std&\_maxexp\_;

 array ses{&\_minexp\_ : &\_maxexp\_} se&\_minexp\_ - se&\_maxexp\_;

 array mdns{&\_minexp\_ : &\_maxexp\_} mdn&\_minexp\_ - mdn&\_maxexp\_;

 array p25s{&\_minexp\_ : &\_maxexp\_} p25&\_minexp\_ - p25&\_maxexp\_;

 array p75s{&\_minexp\_ : &\_maxexp\_} p75&\_minexp\_ - p75&\_maxexp\_;

 %do i=&\_minexp\_ %to &\_maxexp\_; i=&i;

 if &exposure eq i then do;

 ns{i}=n; mns{i}=mean; stds{i}=std; ses{i}=se ;mdns{i}=median; p25s{i}=p25; p75s{i}=p75;

 end;

 %end;

 %do i=&\_minexp\_ %to &\_maxexp\_;

 label n&i="&&label&i n"

 mn&i="&&label&i mean"

 std&i="&&label&i standard dev"

 se&i="&&label&i standard error"

 mdn&i="&&label&i median"

 p25&i="&&label&i p25"

 p75&i="&&label&i p75"

 ;

 %end;

run;

proc means noprint data=\_stdm\_;

var %do i=&\_minexp\_ %to &\_maxexp\_; n&i mn&i std&i se&i mdn&i p25&i p75&i %end;

;

output out=tab mean=;

by ord varname;

run;

%if %str(%upcase(&nortf)) eq %str(T) %then %do;

 options notes;

title5 "Table 1: &header";

%if "&explab" ne "" %then %do;

 title6 "&explab";

%end;

proc print label data=tab noobs; var varname

%do i=&\_minexp\_ %to &\_maxexp\_;

 n&i mn&i std&i se&i mdn&i p25&i p75&i

%end;

;

 run;

 options &notes;

 title5 " ";

%end;

 options notes;

%if "%upcase(&nortf)" = "T" %then %goto exit;

%t1rtf(data = &data,

 exposure = &exposure,

 varlist = &varlist,

 noadj = &noadj,

 cat = &cat,

 mdn = &mdn,

 rtftitle = &rtftitle,

 landscape = &landscape,

 fn = &fn,

 uselbl = &uselbl,

 file=&file,

 dec=&dec,

 dec\_rnd=&dec\_rnd);

goptions reset=all;

proc datasets nolist;

 delete descrips Null

table1dat

dsetcont

frmts

allrows:

tbl1adj

\_tmp\_tbl1\_dat\_

\_tmp\_

\_minmax\_

mainfreq

\_tmp1\_

descrips

stderrsq

\_stdm\_

tab;

run;

%goto exit;

%exit: %mend;

DATA HIV; SET HIV;

LABEL

\_1MRC= 'Multiracial (clnc), 2021 (%)'

\_100PLC = '<100% Poverty Level (clnc), 2021 (%)'

\_1324HIV = '13-24 (cnty), 2020 (%)'

\_1324NH = 'HIV incidence 13-24 (cnty), 2020 (%)'

\_200PLC = '<200% Poverty Level (clnc), 2021 (%)'

\_2534HIV = '25-34 (cnty), 2020 (%)'

\_2534NH = 'HIV incidence 25-34 (cnty), 2020 (%)'

\_3544H = '35-44 (cnty), 2020 (%)'

\_3544NH = 'HIV incidence 35-44 (cnty), 2020 (%)'

\_4554H = '45-54 (cnty), 2020 (%)'

\_4554NH = 'HIV incidence 45-54 (cnty), 2020 (%)'

\_55PH = '>54 (cnty), 2020 (%)'

\_55PNH = 'HIV incidence >54 (cnty), 2020 (%)'

AC = 'Prenatal patients attended 1st tri visit (clnc), 2021 (%)'

AHF = 'Low-income and living far from grocery stores (cnty), 2019 (%)'

Address = 'Clinic address (clnc), 2021'

AWC = '<18 receiving weight screening (clnc), 2021 (%)'

Adult\_Adolescent\_HIV\_Diagnosed = 'Adult adolescent HIV incidence (per 100K) (state), 2020'

ACE= '18-64 (clnc), 2021 (%)'

AWSC = 'Adults receiving weight screening (clnc), 2021 (%)'

AIDS\_Death\_Rate = 'AIDS mortality rate (per 100K) (state), 2019'

APRP = 'AIDS prevalence (state), 2019 (%)'

AIDS\_Rate = 'AIDS incidence (per 100K) (state), 2019'

Air\_Pollution = 'Avg PM2.5 (cnty), 2018'

ADD = 'Driving deaths involving alcohol (cnty), 2016-20 (%)'

AI = 'American Indian (zipc), 2020 (%)'

AMS = 'Female Medicare enrollees 65-74 receiving annual mammogram (cnty), 2019, (%)'

As= 'Asian (zipc), 2020 (%)'

AsC = 'Asian (clnc), 2021, (%)'

AsthC = 'Asthmatic (clnc), %'

ATC = 'Asthmatic patients with treatment plan (clnc), 2021 (%)'

BH = 'Bachelors degree or higher (zipc), 2020 (%)'

BPL = 'Households below poverty level (state), 2019 (%)'

BL= 'Black (zipc), 2020 (%)'

BAAC = 'Black (clnc), 2021 (%)'

BHIV = 'Black (cnty), 2020 (%)'

BPCC = 'Hypertensive with controlled BP (clnc), 2021 (%)'

BPUCC = 'Hypertensive with uncontrolled BP (clnc), 2021 (%)'

BRCC = 'Females receiving mammogram (clnc), 2021 (%)'

BBA = 'Households with broadband internet (cnty), 2016-20 (%)'

C19\_AgeAdj\_Mortality = 'Age-adjusted COVID-19 mortality (per 100K) (cnty), 2020'

CSC = 'Females receiving pap smear (clnc), 2021 (%)'

CCC = 'Median income of households with >1 child spent on childcare (cnty), 2020-21 (%)'

Child\_CareCenters = 'Childcare centers (per 100K aged <5) (cnty), 2021'

Child\_Mortality\_Value = 'Resident <18 mortality (per 100K) (cnty), 2017-20'

CVC= 'Receiving childhood vaccines (clnc), 2021 (%)'

Chil= 'Have children (zipc), 2021 (%)'

ChilC = '<18 (clnc), 2021 (%)'

CFRL = 'Children in public schools eligible for free/reduced lunch (cnty), 2019-20 (%)'

CIP = '<18 and in poverty (cnty), 2020 (%)'

CSP = 'Children living in single-parent household (cnty), 2016-20 (%)'

Chlamydia = 'Chlamydia incidence (per 100K) (cnty), 2019 (%)'

Chlamydia\_Transmit = 'Chlamydia transmission rate (per 100K) (cnty), 2020 (%)'

City\_Center = 'Clinic city location (clnc), 2021'

Comments = 'Other clinic information (clnc), 2021'

Commute = 'Mean commute time to work (min) (zipc), 2020'

Contact\_Information = 'Clinic contact info (clnc), 2021'

Contact\_Name = 'Clinic point of contact (clnc), 2021'

County = 'Clinic county location (clnc), 2021'

County\_Premature\_Death = 'Age-adj YLLs due to premature death (per 100K) (cnty), 2018-20'

CRC = 'Receiving colonoscopy (clnc), 2021 (%)'

DeC = 'Receiving dental care (clnc), 2021 (%)'

DSC = 'Receiving dental sealants (clnc), 2021 (%)'

Dentist\_Ratio = 'Ratio of dentists to population (cnty), 2020'

DRC = 'Depressed and in remission (clnc), 2021 (%)'

DSCC = 'Receiving depression screening (clnc), 2021 (%)'

DC = 'Diabetic (clnc), 2021 (%)'

DP = '>19 diagnosed with diabetes (cnty), 2019 (%)'

D65 = '<65 and disabled (zipc), 2020 (%)'

DIS= 'Disabled (zipc), 2020 (%)'

DY = '16-19 not in school and unemployed (cnty), 2016-20 (%)'

DE= 'Age-adj adults reported binge or heavy drinking (cnty), 2019 (%)'

DAW = 'Workforce driving alone to work (cnty), 2016-20 (%)'

DRAT = 'Ratio dentist to population\*1000 (cnty), 2020'

Drug\_OD\_Death = 'Drug overdose rate (per 100K) (cnty), 2018-20'

EM= 'Employed (zipc), 2020 (%)'

ENC = 'Receiving support services (clnc), 2021 (%)'

Est\_HIV\_Death\_Rate = 'Calculated HIV mortality rate (per 100K) (state), 2020'

Est\_HIV\_Incidence\_Rate = 'Estimated HIV incidence (per 100K) (state), 2019'

EXO = 'Access to space for physical activity (cnty), 2020-21 (%)'

FTI = 'PLWH female infected via IDU (cnty), 2020 (%)'

FTO = 'PLWH female infected via other route (cnty), 2020 (%)'

Family\_Size = 'Mean family size (zipc), 2020'

FE= 'Female (zipc), 2020 (%)'

FEHC = 'PLWH female (cnty), 2020 (%)'

FENH = 'HIV incidence female (cnty), 2020 (%)'

Female\_PrEP\_Need = 'Ratio of 2021 female PrEP users to 2020 female HIV incidence (cnty), 2021'

FPU = 'PrEP users female (cnty), 2020 (%)'

Firearm\_Fatal = 'Firearm mortality (per 100K) (cnty), 2016-20'

FLU = 'FFS Medicare enrollees with 2019 flu vaccination (cnty), 2019 (%)'

FOI = 'Population lacking adequate food access (cnty), 2019 (%)'

FTHC = 'PLWH female infected via heterosexual contact (cnty), 2020 (%)'

Funding\_announcements = 'Funding announcement (clnc), 2021'

Gender\_Pay\_Gap = 'Ratio of female to male median earnings for full-time workers (cnty), 2016-20'

Gonnorhea\_Transmit = 'Gonorrhea transmission rate (per 100K) (cnty), 2020'

GOV = 'Government workers (zipc), 2020 (%)'

Grant\_Expenditure\_Center = 'Total yearly clinic grant expenditures ($100K) (clnc), 2021'

Grants\_Funding\_Provider = 'Clinic funding details (clnc), 2021'

G65 = '>65 (zipc), 2020 (%)'

G100K = 'Total yearly grant expenditures ($100K) (clnc), 2021'

HAB\_Provider\_Site\_Name = 'Clinic name (clnc), 2021'

HAB\_Provider\_Type = 'Clinic type (clnc), 2021'

HAST = 'Ischemic Vascular Disease patients receiving Aspirin (clnc), 2021 (%)'

Health\_Center\_Name = 'Clinic name (clnc), 2021'

HIS= 'Hispanic (zipc), 2020 (%)'

HISH = 'Hispanic (cnty), 2020 (%)'

HISLC = 'Hispanic (clnc), 2021 (%)'

HIVC = 'HIV (clnc), 2021 (%)'

HIV\_Criminalization = 'HIV criminalization laws type (state), 2022'

HIV\_Death\_Rate = 'HIV mortality rate (per 100K) (state), 2019'

HIV\_Diagnosis\_Rate = 'HIV incidence (per 100K) (state), 2019'

HIV\_Prevalence\_Rate = 'HIV prevalence (per 100K) (state), 2019'

HIV\_Rate = 'HIV prevalence >13 (per 100K) (cnty), 2019'

HSCC = 'HIV screening (clnc), 2021 (%)'

HIV\_Viral\_Sup = 'Virally suppressed (state), 2019'

HIVLCC = 'Linked to care (clnc), 2021 (%)'

HTEP = 'Ever tested for HIV (cnty), 2016-17 (%)'

HITYP = 'Tested for HIV, 2016-2017 (cnty), (%)'

HOM= 'Houses that are owner-occupied (cnty), 2016-20 (%)'

Homicides = 'Homicide mortality (per 100K) (cnty), 2014-20'

Hours\_Week = 'Mean work hours per week (zipc), 2020'

HOP = 'Households with 1 or more housing problems (cnty), 2014-18 (%)'

HSC = '>25 with HS diploma (cnty), 2016-20 (%)'

HSG = '9th grade cohort graduating in 4 years (cnty), 2018-19 (%)'

Hub = 'ISCI hub name (clnc), 2021'

HYC = 'Hypertensive (clnc), 2021 (%)'

Income\_Inequality = 'Ratio of 80th to 20th income percentile (cnty), 2016-20'

Income\_Inequality\_GINI = 'Income inequality (GINI) coefficient (0.0-1.0) (cnty), 2020'

Infant\_Mortality = 'Neonatal mortality (per 1K live births) (cnty), 2014-20'

Injury\_Deaths = 'Mortality due to injury (per 100K) (cnty), 2016-20'

Juvenile\_Arrests = 'Delinquency case rate (per 1K minors) (cnty), 2019'

K12 = 'Enrolled in K-12 (zipc), 2020 (%)'

Knowledge\_Status = 'Knowledge of status (state), 2019'

Land\_Area\_mi = 'County land area (mi2) (zipc), 2020'

L18 = '<18 (zipc), 2020 (%)'

L5 = '<5 (zipc), 2020 (%)'

Life\_Expectancy = 'Life expectancy (cnty), 2018-20'

LifeExpectancy\_Birth = 'Life expectancy at birth (state), 2020'

LinkCare\_number = 'Incident HIV - linked to care (state), 2019'

LCP = 'Incident HIV - linked to care (cnty), 2020 (%)'

Living\_Wage = 'Hourly wage needed for basic expenses and taxes for an adult and two children ($) (cnty), 2021'

LC = 'Workers driving alone to work for >30 min (cnty), 2016-20 (%)'

LBW = 'Live births with low birth weight (cnty), 2014-20 (%)'

LBWC = 'Babies delivered with low birth weight (clnc), 2021 (%)'

MHRAT = 'Ratio mental health HCPs to population\*1000 (cnty), 2021'

MHIVC = 'PLWH male (cnty), 2020 (%)'

MNH = 'HIV incidence male (cnty), 2020 (%)'

Male\_PrEP\_Need = 'Ratio of 2021 male PrEP users to 2020 HIV incidence (cnty), 2021'

MPUP = 'PrEP users male (cnty), 2020 (%)'

Math\_Scores = 'Mean 3rd grade math test performance (0-10) (cnty), 2018'

Median\_Household\_Income = 'Median household income ($) (cnty), 2020'

MCC = 'Medicaid/CHIP (clnc), 2021 (%)'

Medicaid\_Expansion = 'Year of Medicaid expansion (state), 2023'

MCE = 'Receiving medical care (clnc), 2021 (%)'

MEC = 'Medicare (clnc), 2021 (%)'

MHC = 'Receiving mental health care (clnc), 2021 (%)'

MIC = 'Racial/ethnic minorities (clnc), 2021 (%)'

MV\_Death = 'Motor vehicle crash death rate (per 100K) (cnty), 2014-20'

NAC = 'American Indian or Alaska Native (clnc), 2021 (%)'

NAHC = 'Native Hawaiian or Other (clnc), 2021 (%)'

NCDC = 'Uncontrolled diabetics (clnc), 2021 (%)'

NM = 'Never married (zipc), 2020 (%)'

NBH = 'HIV incidence Black (cnty), 2020 (%)'

NHH = 'HIV incidence Hispanic (cnty), 2020 (%)'

New\_HIV\_Diagnosis = 'HIV incidence (per 100K) (cnty), 2020'

NWH = 'HIV incidence White (cnty), 2020 (%)'

NHCO = '<65 uninsured (zipc), 2020 (%)'

NHCP = 'Adults with no PCP (state), 2021 (%)'

NH12P = 'Adults not seen HCP in past year due to cost (state), 2021 (%)'

OBE= 'Age-adj adults with obese BMI (cnty), 2019 (%)'

OAC = '>64 (clnc), 2021 (%)'

OIC = '3rd party insurance (clnc), 2021 (%)'

OLC = 'Prefer non-English services (clnc), 2021 (%)'

ORAT = 'Ratio of non-physician PCPs to population\*1000 (cnty), 2021'

O1865 = '18-65 (zipc), 2020 (%)'

PH = 'Native Hawaiian or Pacific Islander (zipc), 2020 (%)'

PCPRAT = 'Ratio PCPs to population\*1000 (cnty), 2019'

Per\_Cap\_income = 'Median county income ($) (zipc), 2020'

PPFH = 'Age-adj adults reported fair or poor health (cnty), 2019 (%)'

PPM = 'Age-adj adults reported >14 days of poor mental health per month (cnty), 2019 (%)'

PPP14 = 'Age-adj adults reported >14 days of poor physical health per month (cnty), 2019 (%)'

PP = 'Poverty (cnty), 2020 (%)'

PUH = 'Housing-insecure (cnty), 2020 (%)'

PerPat\_Cost\_Center = 'Clinic operational costs per patient ($) (clnc), 2021'

PI = 'Age-adj >18 reporting no leisure-time physical activity (cnty), 2019 (%)'

PN\_13\_24 = 'Ratio of 2021 13-24 PrEP users to 2020 HIV incidence (cnty), 2020'

PN\_25\_34 = 'Ratio of 2021 25-34 PrEP users to 2020 HIV incidence (cnty), 2020 '

PN\_35\_44 = 'Ratio of 2021 35-44 PrEP users to 2020 HIV incidence (cnty), 2020'

PN\_45\_54 = 'Ratio of 2021 45-54 PrEP users to 2020 HIV incidence (cnty), 2020'

PN\_55 = 'Ratio of 2021 >54 PrEP users to 2020 HIV incidence (cnty), 2020'

PNPatients\_Delivered\_Center = 'Prenatal deliveries (clnc), 2021'

Poor\_Mental\_health = 'Mean reported days of poor Mental Health in last 30 days (cnty), 2019'

Poor\_Physical\_health = 'Mean reported days of poor Physical Health in last 30 days (cnty), 2019'

P25 = '>25 without HS diploma (state), 2019 (%)'

Pop\_Sq\_Mi = 'Density (persons/mi2) (cnty), 2020'

Population = 'Size (zipc), 2020'

PreMature\_AgeAdj\_Mortality = 'Age-adj mortality for <75 (per 100K) (cnty), 2018-20'

PrenatalPatients\_Center = 'Prenatal (clnc), 2021'

PrEP\_Coverage = 'Prescribed PrEP (state), 2019'

PrEP\_Need = 'Ratio of 2021 PrEP users to 2020 HIV incidence (cnty), 2020'

PU34 = 'PrEP users 25-34 (cnty), 2020 (%)'

PU24 = 'PrEP users 13-24 (cnty), 2020 (%)'

PU44 = 'PrEP users 35-44 (cnty), 2020 (%)'

PU54 = 'PrEP users 45-54 (cnty), 2020 (%)'

PU55 = 'PrEP users >55 (cnty), 2020 (%)'

Prevent\_Hospital\_Stay = 'Hospital stays rate (per 100K Medicare enrollees) (cnty), 2019'

Program\_Parts\_Funding\_Provider = 'Ryan White funding parts (clnc), 2021'

Public\_Transportation = 'Public transit score (0.0-10.0) (zipc), 2018'

Rate\_HIV\_Cases = 'HIV prevalence (per 100K) (cnty), 2020'

Rate\_PrEP\_Use\_100K = 'PrEP use (per 100K) (cnty), 2020'

Reading\_scores = 'Mean 3rd grade reading test performance (0-10) (cnty), 2018'

RC = 'Receiving care (cnty), 2020 (%)'

RCHP = 'HIV prevalence % (cnty), 2020 (%)'

RCHP12 = 'HIV prevalence >13 % (cnty), 2019 (%)'

Receipt\_of\_Care = 'Receiving care (state), 2019'

RP = 'Rural population (state), 2019 (%)'

RPU100KP = 'PrEP use (cnty), 2020 (%)'

School\_Funding\_Adequecy = 'Mean difference between actual and required public school spending per student ($) (cnty), 2019'

School\_Segregation = 'School racial segregation level relative to local distribution (0.0-1.0) (cnty), 2020-21'

Segregation\_Black\_White = 'Racial segregation level (0-100) (cnty), 2016-20'

SHC = 'Households spending at least half of income on housing (cnty), 2016-20 (%)'

SMO= 'Adults current smokers (cnty), 2019 (%)'

Social\_associations = 'Social associations membership rate (per 10K) (cnty), 2019'

Source = 'Clinic data source (clnc), 2021'

Sq\_Mt = 'County land area (m2) (zipc), 2020'

State = 'Clinic state location (clnc), 2021'

State\_Center = 'Clinic state location (clnc), 2021'

STCC = 'Receiving preventive statin therapy for CVD (clnc), 2021 (%)'

SAC = 'Receiving substance use services (clnc), 2021 (%)'

Suicides = ' Age-adj suicide rate (per 100K) (cnty), 2016-20'

Syphilis\_Transmit = 'Syphilis transmission rate (per 100K) (cnty), 2020'

Syringe\_Exchange = 'Syringe exchanges (state), 2022'

Teen\_Births = 'Teen births (per 1K females 15-19) (cnty), 2014-20'

TSC = 'Adults receiving tobacco screening services (clnc), 2021 (%)'

TC100K = 'Total clinic operational costs in last year ($100K) (clnc), 2021'

Total\_Patients\_Center = 'Clinic (clnc), 2021'

Traffic\_Volume = 'Mean major roadway traffic (vol/m) (cnty), 2019'

TMIDU = 'PLWH male infected via IDU (cnty), 2020 (%)'

TMM = 'PLWH male infected via sex with men (cnty), 2020 (%)'

TMH = 'PLWH male infected via heterosexual contact (cnty), 2020 (%)'

TMS = 'PLWH male infected via sex with men and IDU (cnty), 2020 (%)'

TMO = 'PLWH male infected through other means (cnty), 2020 (%)'

TPR = 'Multiracial (zipc), 2020 (%)'

U65NI = '<65 uninsured (cnty), 2019 (%)'

UNE = '>16 unemployed and seeking work (cnty), 2020 (%)'

UA = 'Adults <65 uninsured (cnty), 2019 (%)'

UC = '<19 uninsured (cnty), 2019 (%)'

UCC = 'Uninsured (clnc), 2021 (%)'

UGIS = 'Adults uninsured (state), 2019 (%)'

UCH = 'Uninsured (cnty), 2019 (%)'

Urban\_Suburban\_rural = 'County community type (cnty), 2019'

Urbanicity = 'Urbanicity level (cnty), 2020'

VH = 'Vacant housing (state), 2019 (%)'

Violent\_Crime = 'Violent crime offense rate (per 100K) (cnty), 2014-16'

VS = 'Virally suppressed (cnty), 2020 (%)'

VC = 'Receiving eye care (clnc), 2021 (%)'

Walkability = 'Walkability score (0-100) (zipc), 2022'

Water\_Violations = 'Drinking water violations present (cnty), 2020'

WHI= 'White (zipc), 2020 (%)'

WHH = 'White (cnty), 2020 (%)'

WHO = 'Non-Hispanic White (zipc), 2020 (%)'

Zip = 'Clinic zip code (clnc), 2021';

RUN;

%table1(data = Work.Hiv,

ageadj=F,

noexp=T,

Varlist =

HTEP

HITYP

Knowledge\_Status

LinkCare\_number

LCP

Receipt\_of\_Care

RC

HIV\_Viral\_Sup

VS

PrEP\_Coverage

RPU100KP

FPU

MPUP

PU24

PU34

PU44

PU54

PU55

PrEP\_Need

Female\_PrEP\_Need

Male\_PrEP\_Need

PN\_25\_34

PN\_35\_44

PN\_45\_54

PN\_55

HIV\_Death\_Rate

Est\_HIV\_Death\_Rate

AIDS\_Death\_Rate,

miscol=t, dec =1,

fn = @HIV\_Death\_Rate Mortality is among those who are HIV+ and with AIDS (respectively) and are not necessarily attributed to HIV/AIDS-related causes,

rtftitle= Implementation Outcome Variables (Mean (SD) or % (SD), and % Missing),

file = IOVariableTableDraft);

run;

%table1(data = Work.Hiv,

ageadj=F,

noexp=T,

varlist =

Population

FE

Pop\_Sq\_Mi

RP

Urban\_Suburban\_rural

Urbanicity

L5

L18

O1865

G65

As

WHI

WHO

BL

AI

PH

HIS

TPR

Family\_Size

Chil

CCC

CSP

NM

Social\_associations

Teen\_Births

Per\_Cap\_income

Median\_Household\_Income

Gender\_Pay\_Gap

Income\_Inequality

Income\_Inequality\_GINI

Living\_Wage

PP

CIP

BPL

AHF

HOP

PUH

HOM

SHC

VH

EM

Hours\_Week

GOV

UNE

DY

CFRL

K12

HSG

HSC

P25

BH

Reading\_scores

Math\_Scores

School\_Funding\_Adequecy

School\_Segregation

Land\_Area\_mi

Sq\_Mt

Segregation\_Black\_White

Child\_CareCenters

BBA

Air\_Pollution

FOI

EXO

Commute

LC

DAW

MV\_Death

Traffic\_Volume

Public\_Transportation

Walkability

Violent\_Crime

Homicides

Juvenile\_Arrests

Water\_Violations

DE

NHCP

NH12P

PCPRAT

DRAT

MHRAT

ORAT

NHCO

U65NI

UA

UC

UGIS

UCH

Syringe\_Exchange

Medicaid\_Expansion,

miscol=t, dec =1,

poly = Urban\_Suburban\_rural Water\_Violations Urbanicity,

rtftitle= Environmental Variables (Mean (SD) or % (SD), and % Missing),

file = EnvVarTable);

run;

%table1(data = Work.Hiv,

ageadj=F,

noexp=T,

varlist =

Total\_Patients\_Center

ChilC

ACE

OAC

MIC

AsC

BAAC

HISLC

NAC

NAHC

\_1MRC

OLC

HIVC

HIVLCC

PrenatalPatients\_Center

AC

PNPatients\_Delivered\_Center

LBWC

DC

NCDC

HYC

BPUCC

DRC

AsthC

MCE

STCC

DeC

DSC

MHC

SAC

VC

ENC

HAST

AWSC

AWC

BRCC

CSC

CRC

DSCC

TSC

\_100PLC

\_200PLC

MEC

MCC

OIC

UCC

TC100K

G100K

PerPat\_Cost\_Center,

miscol=t, dec =1,

fn = All Missingness due to many clinics not being included in the HRSA database which contained the information on clinic-level variables,

rtftitle= Clinic Variables (Mean (SD) or % (SD), and % Missing),

file = ClinicVariableTableDraft);

run;

%table1(data = Work.Hiv,

ageadj=F,

noexp=T,

varlist =

HIV\_Prevalence\_Rate

Rate\_HIV\_Cases

RCHP

HIV\_Rate

RCHP12

\_1324HIV

\_2534HIV

\_3544H

\_4554H

\_55PH

BHIV

WHH

HISH

HIV\_Diagnosis\_Rate

Est\_HIV\_Incidence\_Rate

New\_HIV\_Diagnosis

MNH

FENH

NBH

NWH

NHH

Adult\_Adolescent\_HIV\_Diagnosed

\_1324NH

\_2534NH

\_3544NH

\_4554NH

\_55PNH

MHIVC

TMH

TMIDU

TMM

TMS

TMO

FTI

FTHC

FTO

AIDS\_Rate

APRP

HIV\_Criminalization,

miscol=t, dec =1,

poly = HIV\_Criminalization,

fn = @Est\_HIV\_Incidence\_Rate Estimated time of infection using CD4 depletion model from CDC ,

rtftitle= HIV Variables (Mean (SD) or % (SD), and % Missing),

file = HIVTableDraft);

run;