



Transforming Statistical Programming with Code Traceability as the Information Infrastructure for AI



Tomás Sabat Stöfsei

CEO & Co-Founder, Verisian

Build your **clinical trial AI** organization with Verisian

Today

Today

Define-xml: auto-generated, derivation summaries for every custom-derived variables

Name	Label	Description
TRTEMFL	Treatment Emergent Flag	TRTEMFL, or treatment-emergent flag, is set for any adverse event arising or getting worse after the first
TRTEMFL	Treatment Emergent Flag	TRTEMFL, or treatment-emergent flag, is set for any adverse event arising or getting worse after the first
TRTEMFL	Treatment	

Analysis Data Reviewer’s Guide

ADTTE - Time-to-Event Analysis Data

Inputs	Datasets	Outputs	TLFs
	ADSL		
	ADVS		
	ADQS		
	ADCM		

The ADTTE dataset is assembled by integrating randomization and treatment start dates from the ADaM adsl dataset, pain intensity scores and analgesic use from the ADaM advs and adqs datasets, medication-start information from adcm ...

Today

Define-xml: auto-generated, derivation summaries for every custom-derived variables

Reviewer's Guides: auto-generated dataset summaries, full dataset lineage

Name	Label	Description
TRTEMFL	Treatment Emergent Flag	TRTEMFL, or treatment-emergent flag, is set for any adverse event arising or getting worse after the first
TRTEMFL	Treatment Emergent Flag	TRTEMFL, or treatment-emergent flag, is set for any adverse event arising or getting worse after the first
TRTEMFL	Treatment	

Analysis Data Reviewer's Guide

ADTTE - Time-to-Event Analysis Data

Inputs	Datasets	Outputs	TLFs
	ADSL		
	ADVS		
	ADQS		
	ADCM		

The ADTTE dataset is assembled by integrating randomization and treatment start dates from the ADaM adsl dataset, pain intensity scores and analgesic use from the ADaM advs and adqs datasets, medication-start information from adcm ...

Today

Define-xml: auto-generated, derivation summaries for every custom-derived variables

Reviewer's Guides: auto-generated dataset summaries, full dataset lineage

A lot more to come, stay tuned...

Name	Label	Description
TRTEMFL	Treatment Emergent Flag	TRTEMFL, or treatment-emergent flag, is set for any adverse event arising or getting worse after the first
TRTEMFL	Treatment Emergent Flag	TRTEMFL, or treatment-emergent flag, is set for any adverse event arising or getting worse after the first
TRTEMFL	Treatment	

Analysis Data Reviewer's Guide

ADTTE - Time-to-Event Analysis Data

Inputs	Datasets	Outputs	TLFs
	ADSL		
	ADVS		
	ADQS		
	ADCM		
			Cox proportional hazards model used to derive the hazard ratio for the treatment group effect

The ADTTE dataset is assembled by integrating randomization and treatment start dates from the ADaM adsl dataset, pain intensity scores and analgesic use from the ADaM advs and adqs datasets, medication-start information from adcm ...

Make your studies AI-ready in **one day**



Verisian

AI-ready information
infrastructure for clinical trials

Statisticians

Clinicians

Data managers

Medical writers

Statistical programmers

...

↻ File transfer

Data, TLFs, Code,
SAP, specs



Your systems



SCE



Storage



...

Across any **operating model**



In-house teams

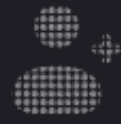


CRO outsourcing

...the trial analysis lifecycle



Planning



Onboarding



Programming



Validation



Submission



Regulatory review

...and your entire study team



Perfect transparency, trust and confidence in submissions

Perfect transparency, trust and confidence in submissions

Handle more studies, faster, with fewer resources at higher quality

Perfect transparency, trust and confidence in submissions

Handle more studies, faster, with fewer resources at higher quality

Lower cost

Introduce a **paradigm shift**

Code Traceability

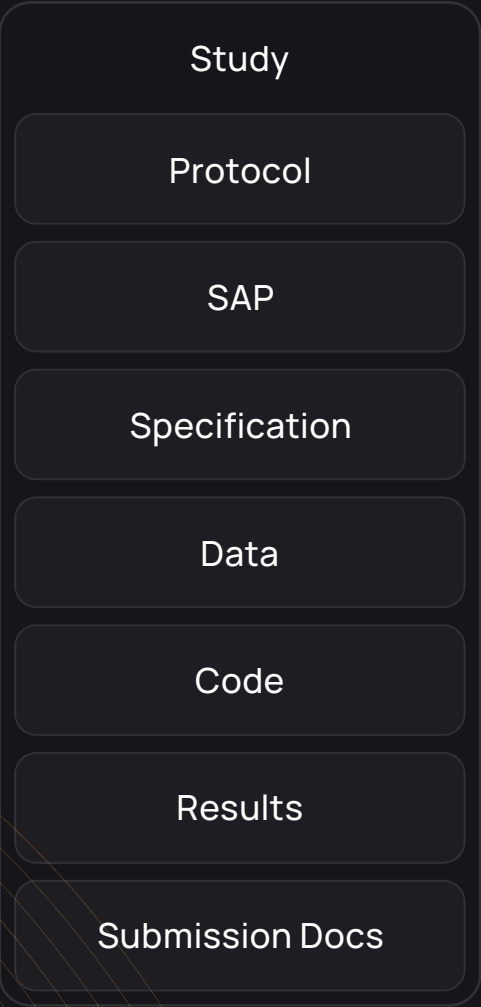
Connects your specifications, data, code, results (TLFs), and submission documentation

Code Traceability

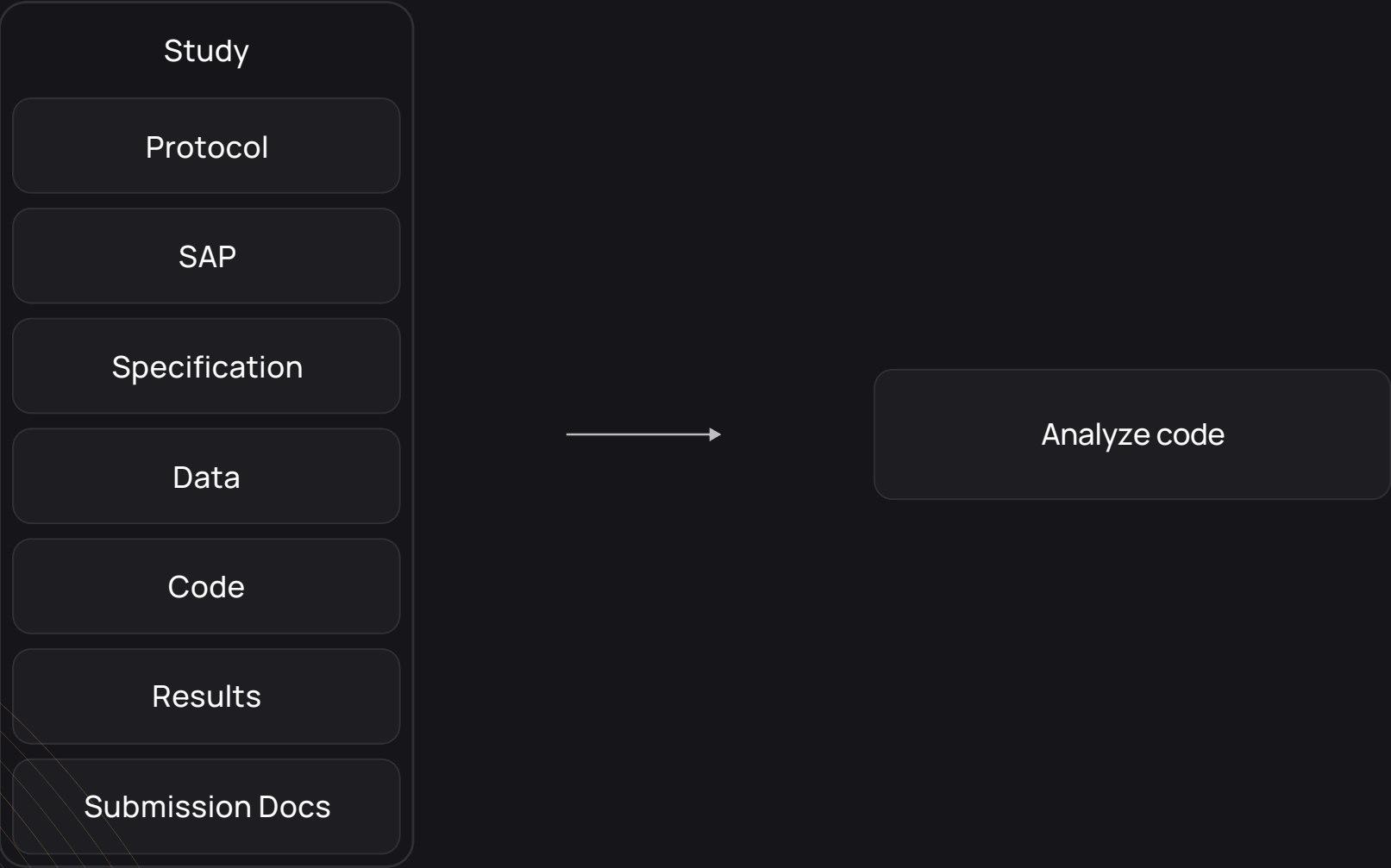
Connects your specifications, data, code, results (TLFs), and submission documentation

Independent of CDISC standards, language (R or SAS), and SCEs

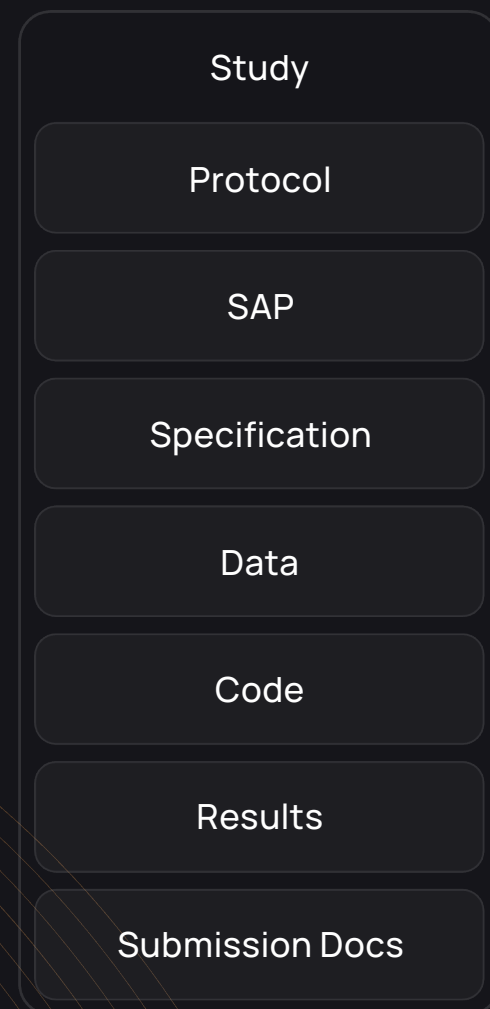
Building Traceability: From Code to Graph



Building Traceability: From Code to Graph



Building Traceability: From Code to Graph



Analyze code



The **source of truth** of a clinical trial is its **code**

Variable Traceability: SDTM.DM.AGE in a demo study

AGE is a simple concept, but can be derived in different ways

Variable Traceability: SDTM.DM.AGE in a demo study

AGE is a simple concept, but can be derived in different ways

dob	firstdose
1974-02-05	2010-03-28
...	...
...	...
...	...
...	...

Variable Traceability: SDTM.DM.AGE in a demo study

AGE is a simple concept, but can be derived in different ways

dob	firstdose
1974-02-05	2010-03-28
...	...
...	...
...	...
...	...



```
age.sas
...
firstdose=min(firstdose,startdt,enddt);
age=intck('year',dob,firstdose,'c');
```

Variable Traceability: SDTM.DM.AGE in a demo study

AGE is a simple concept, but can be derived in different ways

dob	firstdose
1974-02-05	2010-03-28
...	...
...	...
...	...
...	...



```
age.sas
...
firstdose=min(firstdose,startdt,enddt);
age=intck('year',dob,firstdose,'c');
```



age
36
...
...
...
...

How is **AGE** derived and what data does it depend on?

 macros	 source	 sdtm	 adam	 tlf
 %derive_baseline.sas	 demographic.sas	 dm.sas	 adae.sas	 t14_1_1.sas
 %dtc2dt.sas	 dosing.sas	 ae.sas	 adef.sas	 t14_2_1.sas
 %derive_age.sas	 labs.sas	 cm.sas	 adsl.sas	 t14_3_1.sas
 %flag_last.sas	 adverse.sas	 ex.sas	 adef.sas	 l162_1.sas
 %km_plot.sas	 exposure.sas	 lb.sas	 adtte.sas	 l16_2_2.sas
 %visit_window.sas	 vital_signs.sas	 vs.sas	 advs.sas	 f14_2_1.sas
 +14 more	 +18 more	 +30 more	 +16 more	 +174 more

dm.sas

```
proc sort data=source.dosing(keep=subject startdt enddt) out=dosing;
by subject startdt;
run;

data dose_sum;
set dose_raw;
by subject;

retain firstdose lastdose;

if first.subject then do;
firstdose=.;
lastdose=.;
end;

firstdose=min(firstdose,startdt,enddt);
lastdose=max(lastdose,startdt,enddt);

if last.subject;
run;

proc sort data=source.demographic out=demographic;
by subject;
run;

data demog_dose;
merge demographic dosing;
by subject;
run;

options missing=' ';

data sdtm.dm;
set EMPTY_DM demog_dose(rename=(race=_race));

studyid='XYZ123';
domain='DM';
usubjid=left(uniqueid);
subjid=put(subject,3.);
rfstdtc=put(firstdose,yymmdd10.);
rfendtc=put(lastdose,yymmdd10.);
rfxstdtc=put(firstdose,yymmdd10.);
rfxendtc=put(lastdose,yymmdd10.);
rficdte=put(icdate,yymmdd10.);
rfpendtc=put(lastdoc,yymmdd10.);
dthfl='N';
siteid=substr(subjid,1,1) || "00";
brthdte=put(dob,yymmdd10.);
age=intck('year', dob, firstdose, 'c');
if age ne . then ageu='YEARS';
sex=put(gender,sex_demographic_gender.);
race=put(_race,race_demographic_race.);
armcd=put(trt,armcd_demographic_trt.);
arm=put(trt,arm_demographic_trt.);
actarmcd=put(trt,armcd_demographic_trt.);
actarm=put(trt,arm_demographic_trt.);
country="USA";
run;
```

dm.sas

```
proc sort data=source.dosing(keep=subject startdt enddt) out=dosing;
by subject startdt;
run;

data dose_sum;
set dose_raw;
by subject;

retain firstdose lastdose;

if first.subject then do;
firstdose=.;
lastdose=.;
end;

firstdose=min(firstdose,startdt,enddt);
lastdose=max(lastdose,startdt,enddt);

if last.subject;
run;

proc sort data=source.demographic out=demographic;
by subject;
run;

data demog_dose;
merge demographic dosing;
by subject;
run;

options missing=' ';

data sdm.dm;
set EMPTY_DM demog_dose(rename={race=_race});

studyid='XYZ123';
domain='DM';
usubjid=left(uniqueid);
subjid=put(subject,3.);
rfstdtc=put(firstdose,ymmddl0.);
rfendtc=put(lastdose,ymmddl0.);
rfxstdtc=put(firstdose,ymmddl0.);
rfxendtc=put(lastdose,ymmddl0.);
rficdte=put(icdate,ymmddl0.);
rfpendtc=put(lastdoc,ymmddl0.);
dthfl='N';
siteid=substr(subjid,1,1) || "00";

age=intck('year', dob, firstdose, 'c');

sex=put(gender,sex_demographic_gender.);
race=put(_race,race_demographic_race.);
armcd=put(trt,armcd_demographic_trt.);
arm=put(trt,arm_demographic_trt.);
actarmcd=put(trt,armcd_demographic_trt.);
actarm=put(trt,arm_demographic_trt.);
country="USA";
run;
```


How is AGE derived and what data does it depend on?

macros	source	sdtm	adam	tlf
 %derive_baseline.sas	 demographic.sas	 dm.sas	 adae.sas	 t14_1_1.sas
 %dtc2dt.sas	 dosing.sas	 ae.sas	 adef.sas	 t14_2_1.sas
 %derive_age.sas	 labs.sas	 cm.sas	 adsl.sas	 t14_3_1.sas
 %flag_last.sas	 adverse.sas	 ex.sas	 adef.sas	 l162_1.sas
 %km_plot.sas	 exposure.sas	 lb.sas	 adtte.sas	 l16_2_2.sas
 %visit_window.sas	 vital_signs.sas	 vs.sas	 advs.sas	 f14_2_1.sas
 +14 more	 +18 more	 +30 more	 +16 more	 +174 more

```
demographics.sas

proc format;
value trt 1="Active" 0="Placebo";
value gender 1="Male" 2="Female";
value race 1="White" 2="Black" 3="Other";
run;

data source.demographic;
set raw.demographic;
label subject="Subject Number" trt="Treatment" gender="Gender"
race="Race"
orace="Other Race Specify" dob="Date of Birth" uniqueid=
"Company Wide Subject ID" randdt="Randomization Date" icdate=
"Informed Consent Date" lastdoc="Last date of contact";
input subject 1-3 trt 5 gender 7 race 9 orace $ 11-20 +1 dob
mmddyy10. +1
randdt mmddyy10. +1 icdate mmddyy10. +1 lastdoc mmddyy10.;
uniqueid='UNI' || put(subject,3.);
format dob randdt icdate lastdoc mmddyy10.;
run;

data _null_;
rc = filename('xptdir', "&xptdir");
if fexist('xptdir') = 0 then rc = dcreate("xpt",
"/mnt/viya-share/homes/demo-study/data/xpt");
run;

libname xpt_demo xport "&xptdir./demographic.xpt";
data xpt_demo.demograp;
set source.demographic;
run;
libname xpt_demo clear;
```

```
dosing.sas

* dosing.sas;
data source.dosing;
label subject="Subject Number" startdt="Dosing start date" enddt=
"Dosing end date" ddose="Daily dose taken (pills)" uniqueid=
"Company Wide Subject ID" startmm="Month of Start Dose" startdd=
"Day of Start Dose" startyy="Year of Start Dose" endmm=
"Month of End Dose" enddd="Day of End Dose" endyy=
"Year of End Dose";
input subject 1-3 startmm 5-6 startdd 8-9 startyy 11-14 endmm 16-17
enddd
19-20 endyy 22-25 ddose 27;
uniqueid='UNI' || put(subject,3.);
startdt=mdy(startmm, startdd, startyy);
enddt=mdy(endmm, enddd, endyy);
format startdt enddt mmddyy10.;
datalines;
;
run;
```

```
dm.sas

proc sort data=source.dosing(keep=subject startdt enddt) out=dosing;
by subject startdt;
run;

data dose_sum;
set dose_raw;
by subject;

retain firstdose lastdose;

if first.subject then do;
firstdose=.;
lastdose=.;
end;

firstdose=min(firstdose,startdt,enddt);
lastdose=max(lastdose,startdt,enddt);

if last.subject;
run;

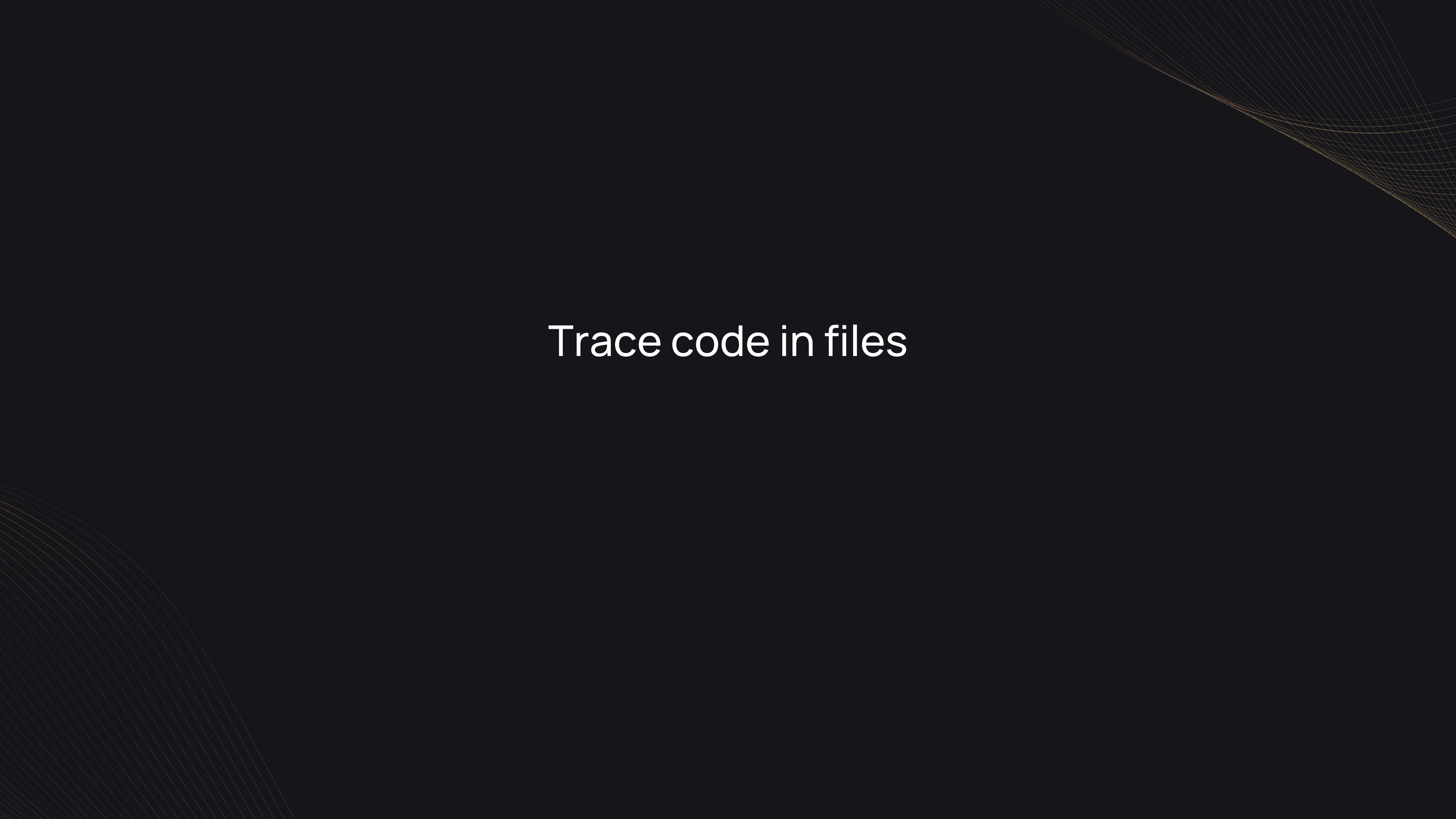
proc sort data=source.demographic out=demographic;
by subject;
run;

data demog_dose;
merge demographic dosing;
by subject;
run;

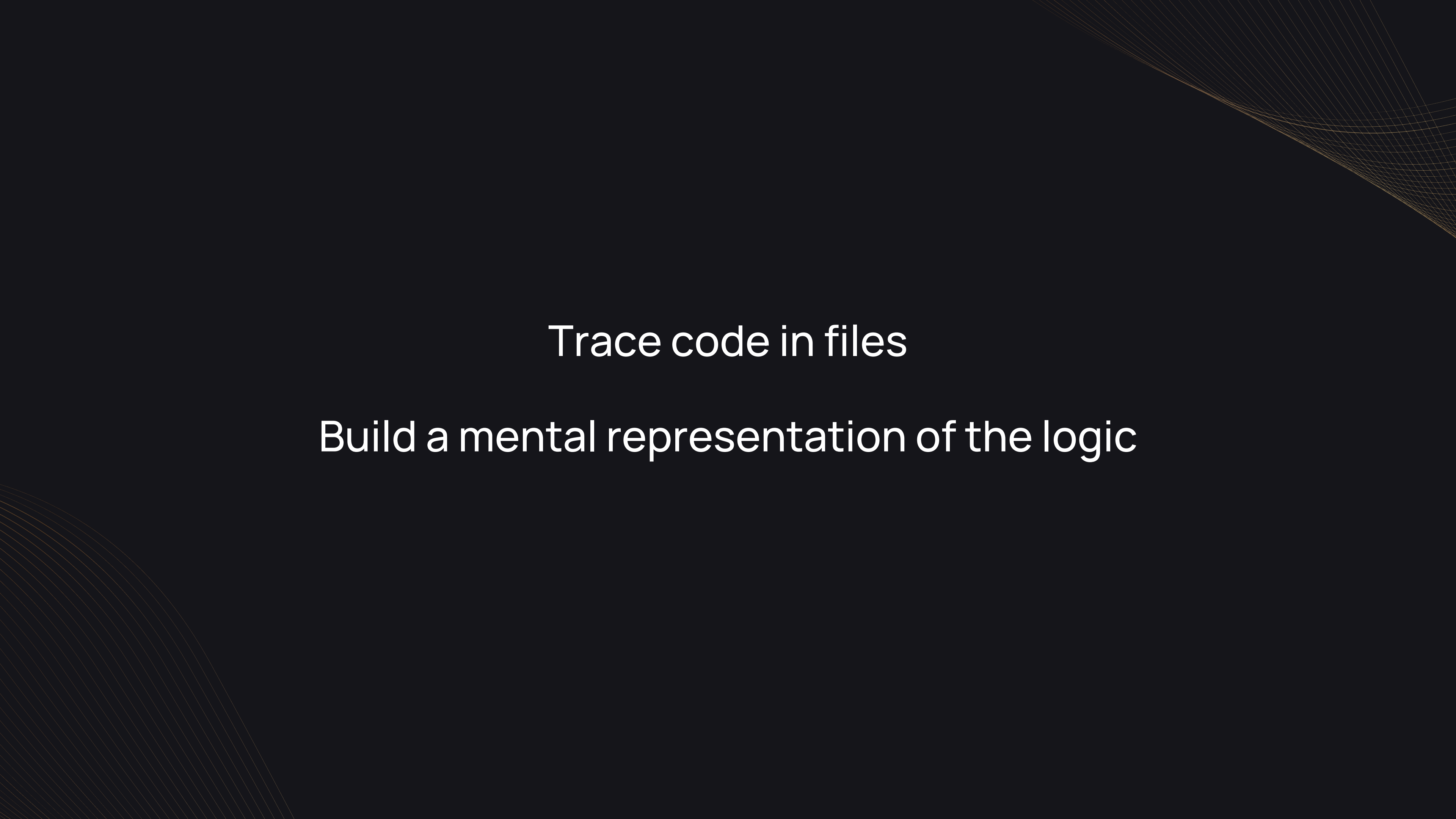
options missing=' ';

data sdtm.dm;
set EMPTY_DM demog_dose(rename=(race=_race));

studyid='XYZ123';
domain='DM';
usubjid=left(uniqueid);
subjid=put(subject,3.);
rfstdtc=put(firstdose,ymmdd10.);
rfendtc=put(lastdose,ymmdd10.);
rfxstdtc=put(firstdose,ymmdd10.);
rfxendtc=put(lastdose,ymmdd10.);
rficdtc=put(icdate,ymmdd10.);
rfpendtc=put(lastdoc,ymmdd10.);
dthfl='N';
siteid=substr(subjid,1,1) || "00";
brthdte=put(dob,ymmdd10.);
age=intck('year', dob, firstdose, 'c');
if age ne . then ageu='YEARS';
sex=put(gender,sex_demographic_gender.);
race=put(_race,race_demographic_race.);
armcd=put(trt,armcd_demographic_trt.);
arm=put(trt,arm_demographic_trt.);
actarmcd=put(trt,armcd_demographic_trt.);
actarm=put(trt,arm_demographic_trt.);
country="USA";
run;
```

Trace code in files



Trace code in files

Build a mental representation of the logic

What variables does **AGE** depend on?

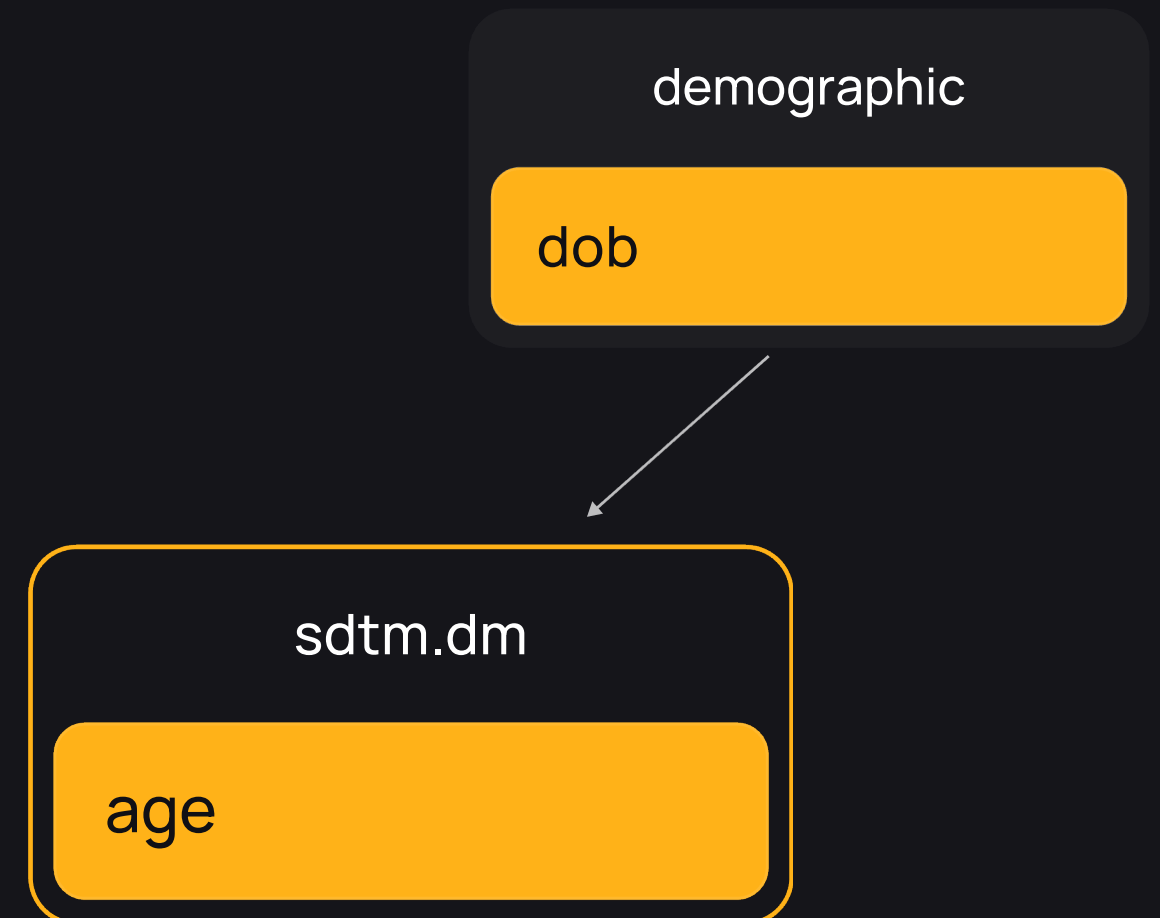
```
data sdtm.dm;  
age=intck('year',dob,firstdose,'c');
```

sdtm.dm

age

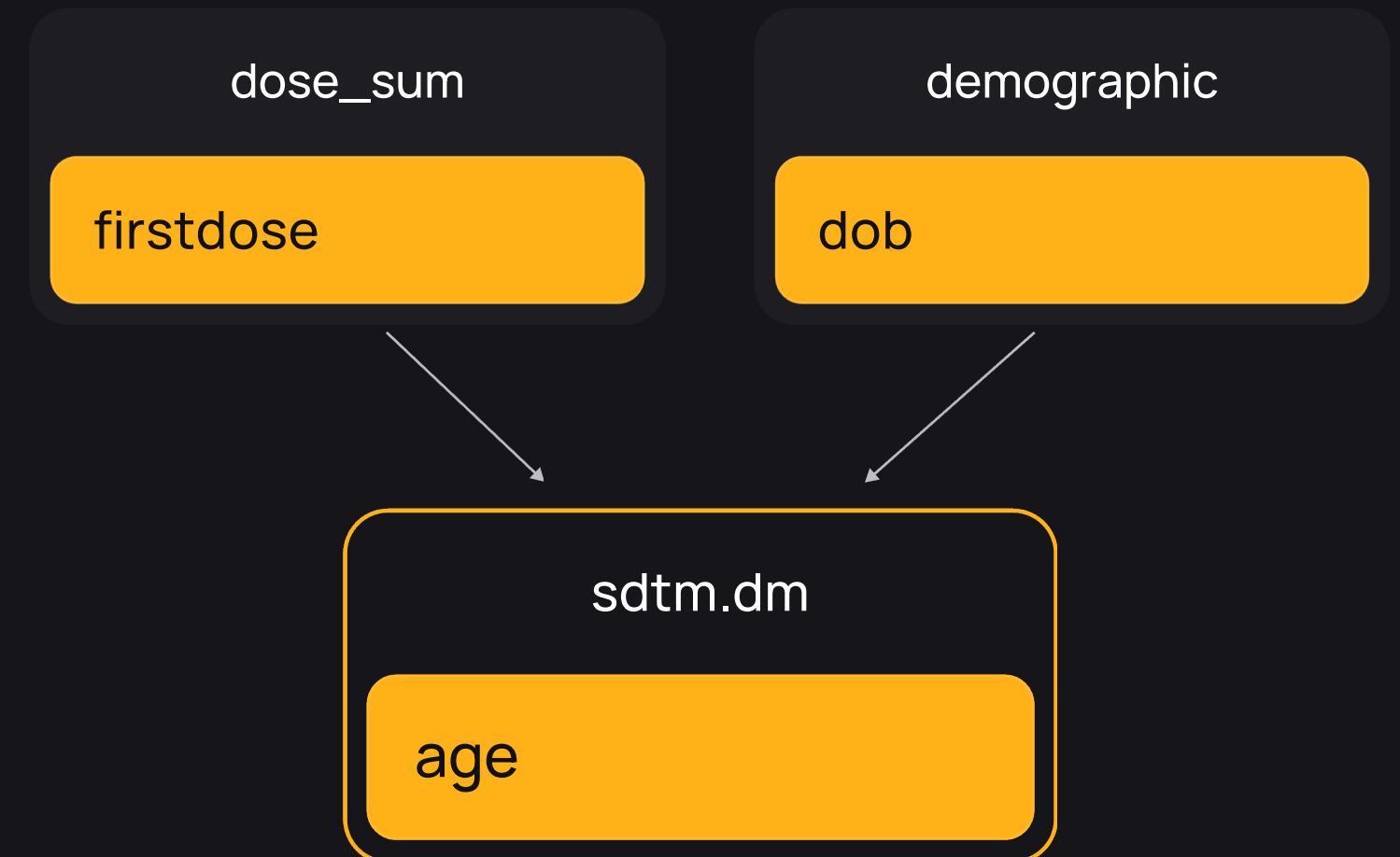
What variables does **AGE** depend on?

```
data demographic;  
set raw_demographics;  
  
data sdtm.dm;  
set demographic dose_sum;  
age=intck('year', dob, firstdose, 'c');
```



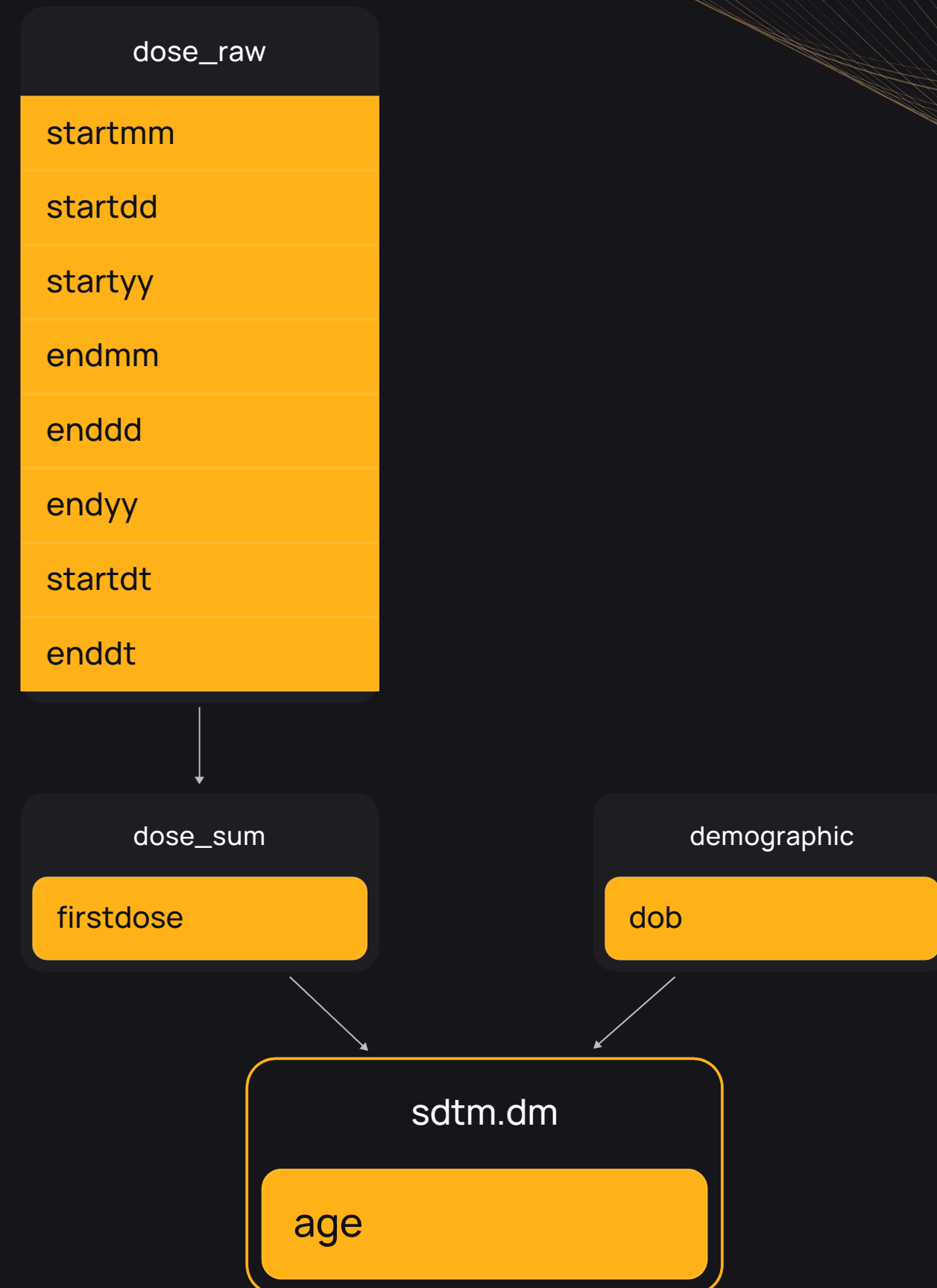
What variables does **AGE** depend on?

```
data dose_sum;  
if first.subject then do;  
firstdose=.;  
firstdose=min(firstdose, startdt, enddt);  
  
data demographic;  
set raw_demographics;  
  
data sdtm.dm;  
set demographic dose_sum;  
age=intck('year', dob, firstdose, 'c');
```



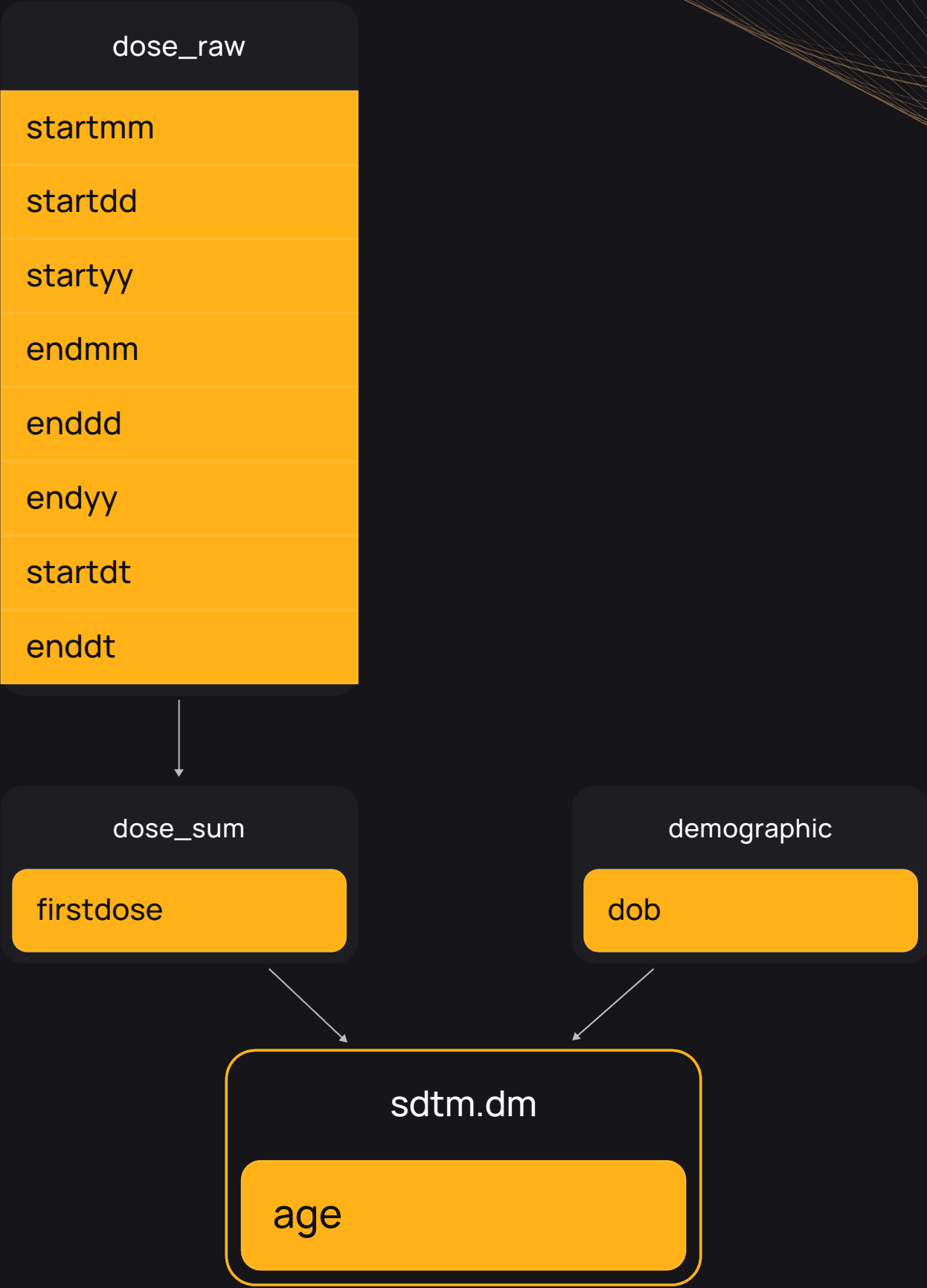
What variables does **AGE** depend on?

```
data dose_raw;  
startdt=mdy(startmm, startdd , startyy);  
enddt=mdy(endmm, enddd, endyy);  
run;  
  
data dose_sum;  
if first.subject then do;  
firstdose=.;  
firstdose=min(firstdose,startdt,enddt);  
run;  
  
data demographic;  
set raw_demographics;  
run;  
  
data sdtm.dm;  
set demographic dose_sum;  
age=intck('year', dob, firstdose, 'c');
```



What variables does AGE depend on?

Dataset	Variable
dose_raw	startmm
dose_raw	startdd
dose_raw	startyy
dose_raw	endmm
dose_raw	enddd
dose_raw	endyy
dose_raw	startdt
dose_raw	enddt
dose_raw	firstdose
demographic	dob



What is **age**?

dob	firstdose
1974-02-05	2010-03-28
...	...
...	...
...	...
...	...



```
age.sas
...
firstdose=min(firstdose,startdt,enddt);
age=intck('year',dob,firstdose,'c');
```



age
36
...
...
...
...

Validating results

Advanced Validation: **Study Integrity**

- Example: CNSR is crucial in time-to-event analyses and TLFs, including Kaplan-Meier analyses
- Task: Ensure CNSR is used appropriately for TTE TLFs

 The variable **cnsr** impacts 1 result (TLFs).

 Search TLFs

Title  

File  

Trace

Cox proportional hazards model used to derive the hazard ratio for the treatment group effect

--



Results validation14> 1> Kaplan-Meier estimates and log-rank test for the treatment effect

PDFs

Table 14.4
Kaplan-Meier estimates and log-rank test for the treatment effect
Full Analysis Set

	Analgezia (N=27)	Placebo (N=27)
Number of events	37 (46.8%)	39 (48.1%)
Number censored	42(26.6%)	42(17.3%)
Median Time to Pain Relief (days)	96.0	94.0
95% CI	[93.0,187.0]	[93.0,98.0]
P-value	0.7488	
Hazard ratio	0.93	
95% CI	[0.59,1.46]	

Step

Derivation

Resolved

Original

Trace health

```
1 * tlf-lifetest.sas;
2 proc report data=kapmei nowd headline headskip
3   style(report)=[rules=groups cellspacing=3 cellpadding=1]
4   style(header)=[just=c background=white];
5
6   columns param trt1 trt2;
7
8   define param / display " " style(column)=[cellwidth=3.1in just=l];
9   define trt1 / display "Analgezia^n(N=27)" style(column)=[cellwidth=1.8in just=c];
10  define trt2 / display "Placebo^n(N=27)" style(column)=[cellwidth=1.8in just=c];
11
12  /* TOPLINE */
13  compute before _page_ / style=[bordertopwidth=0.75pt bordertopcolor=black];
14  | line ' ';
15  endcomp;
16
17  /* add blank spacing after the "Number censored" row and after "P-value" row */
18  compute after;
19  | if upcase(param) in ('NUMBER CENSORED','P-VALUE') then do;
20  |   line ' ';
21  | end;
22  endcomp;
23
24 run;
```

```

by STUDYID USUBJID XPTSTCD VISITNUM;
run;

*adtl.sas;
%ofb(indatadtm.xp, outdatadtmresponders, dayvar=apdy, avalvar=xpstren,
keepvar=usubjid visitnum chg)

*adtl.sas;
data ADGL;
merge EMPTY_ADGL_DM (in=inDM) responders (in=inresp where=(visitnum>1)) ;
by usubjid;

%convert RFXSTDTC to a numeric SAS date named TRTSDT;
/* ***** version: macro invocation start of <dtc2dt> *****
*
*dtc2dt(RFXSTDTC, prefix=TRTS)
*
*****
if length(trim(RFXSTDTC))=10 and index(RFXSTDTC,'--')=0 then TRTSDtc= input(RFXSTDTC, yymmdd10.);
else if length(RFXSTDTC)=16 and index(RFXSTDTC,'--')=0 and index(RFXSTDTC,':')=0 then do;
TRTSDtc=input(trim(RFXSTDTC)||"--00", a0001d19.);
TRTSDtc=datepart(TRTSDtc);
* optionally add formats: ;
format TRTSDtc datetime16.;
end;
* optionally add formats: ;
format TRTSDtc yymmdd10.;
/* ***** version: macro invocation end of <dtc2dt> *****
*****
/
2
* create BIRTHDT, RANDDT, TRTEDT;
/* ***** version: macro invocation start of <dtc2dt> *****
*
dtc2dt(BIRTHDT, prefix=BIRTH)
*
*****
if length(trim(BIRTHDT))=10 and index(BIRTHDT,'--')=0 then BIRTHdt= input(BIRTHDT, yymmdd10.);
else if length(BIRTHDT)=16 and index(BIRTHDT,'--')=0 and index(BIRTHDT,':')=0 then do;
BIRTHdt=input(trim(BIRTHDT)||"--00", a0001d19.);
BIRTHdt=datepart(BIRTHdt);
* optionally add formats: ;
format BIRTHdt datetime16.;
end;
* optionally add formats: ;
format BIRTHdt yymmdd10.;
/* ***** version: macro invocation end of <dtc2dt> *****
*****
/
2
/* ***** version: macro invocation start of <dtc2dt> *****
*
dtc2dt(RANDTC, prefix=RAND)
*
*****
if length(trim(RANDTC))=10 and index(RANDTC,'--')=0 then RANDdt= input(RANDTC, yymmdd10.);
else if length(RANDTC)=16 and index(RANDTC,'--')=0 and index(RANDTC,':')=0 then do;
RANDdt=input(trim(RANDTC)||"--00", a0001d19.);
RANDdt=datepart(RANDdt);
* optionally add formats: ;
format RANDdt datetime16.;
end;
* optionally add formats: ;
format RANDdt yymmdd10.;
/* ***** version: macro invocation end of <dtc2dt> *****
*****
/
2
/* ***** version: macro invocation start of <dtc2dt> *****
*
dtc2dt(RFXENDTC, prefix=TRTE)
*
*****
if length(trim(RFXENDTC))=10 and index(RFXENDTC,'--')=0 then TRTEDtc= input(RFXENDTC, yymmdd10.);
else if length(RFXENDTC)=16 and index(RFXENDTC,'--')=0 and index(RFXENDTC,':')=0 then do;
TRTEDtc=input(trim(RFXENDTC)||"--00", a0001d19.);
TRTEDtc=datepart(TRTEDtc);
* optionally add formats: ;
format TRTEDtc datetime16.;
end;
* optionally add formats: ;
format TRTEDtc yymmdd10.;
/* ***** version: macro invocation end of <dtc2dt> *****
*****
/
2
/* ***** version: macro invocation start of <dtc2dt> *****
*
dtc2dt(RFXENDTC, prefix=TRTE)
*
*****
*****
/
2
data xp;
set EMPTY_XP source.pain;
studyid="KY2123";
domain="XP";
usubjid=left(uniqueid);

xpstcd="Pain Score";
xpstcds="XPPAIN";
epoch="TREATMENT";

**** transpose pain data;
array dates {3} randomdt month3dt month4dt;
array scores {3} painbase pain3mo pain6mo;

do i=1 to 3;
visitnum=i - 1;
visit=put(visitnum, visit_labr_month.);
if visit="Baseline" then xpb1fl="Y";
else xpb1fl=" ";

if scores[i] ne . then do;
xpcrcs=left( _put(scores[i], pain.));
xptres=putres;
xptresnscores[i];
xpdt=put(dates[i], yymmdd10.);
output;
end;

```

```

if length(trim(xpdt))=10 and index(xpdt,'--')=0 then adtc= input(xpdt, yymmdd10.);
else if length(xpdt)=16 and index(xpdt,'--')=0 and index(xpdt,':')=0 then do;
adtc= input(trim(xpdt)||"--00", a0001d19.);
adtc=datepart(adtc);
* optionally add formats: ;
format adtc datetime16.;
end;
if (.<adt=randtc then ady=adt - randdt;
else if adt=randtc then ady=adt - randdt + 1;
* optionally add formats: ;
format adt yymmdd10.;
/* ***** version: macro invocation end of <dtc2dt> *****
*****
/
2
run;

* adtc.sas;
%ofb(indatadef, outdatadef, dayvar=ady, avalvar=xpstren)

* adtc.sas;
%make_sort_order(metadatafile=mt/viya-share/homes/tomas-2/demo-study/metadata/chapter06/
ADAM_METADATA.xls, dataset=AGEF)

* adtc.sas;
proc sort data=adef(keep=STUDYID USUBJID SITEID COUNTRY RANDDT AGE AGEGR1N AGEGR1 SEX RACE TRTPN TRTP
PARAMCD PARAM AVISITN AVISITN ABLFL XPSQ VISITNUM ADT ADY AVAIL AVAILC BASE CHG CRIT1PL CRIT1 ITTFL)
out=adtc.adef;
by STUDYID USUBJID PARAMCD AVISITN ADT;
run;

* adtc.sas;
proc sort data=adtc.adef(keep=usubjid paramcd chg adt visitnum xpsq) out=adef;
where paramcd="XPPAIN" and visitnum>0 and abs(chg)>0;
by usubjid adt;
run;

* adtc.sas;
data adtc;
set adef;
by usubjid adt;
drop paramcd visitnum;
if first.usubjid;
run;

* adtc.sas;
%make_empty_dataset(metadatafile=mt/viya-share/homes/tomas-2/demo-study/metadata/chapter06/
ADAM_METADATA.xls, dataset=ADAE)

* AE.sas;
%make_empty_dataset(metadatafile=mt/viya-share/homes/tomas-2/demo-study/metadata/chapter02/
SDTM_METADATA.xls, dataset=AE)

* adverse.sas;
data source.adverse;
label subject="Subject Number" bodysys="Body System of Event" preftecm="Preferred Term for Event"
aerel="Relatedness: 1=not, 2=possibly, 3=probably"
aesev="Severity/Intensity: 1=ld, 2=moderate, 3=severe"
aaction="Action taken: 1=drug stopped, 2=dose reduced, 3=dose increased, 4=no dose change, 5=unknown"
aestart="AE Start date" aend="AE End date" serious="Serious AEP" aetxt="Event Verbatim Text"
uniqueid="Company Wide Subject ID";
infile datalines linesize=324 truncover;
input subject aerel aesev aaction aestart mddy10. -1 aend mddy10.
bodysys $ 30-60 prefcm $ 60-90 aerel $ 90-110 aesev $ 120-128 aaction $
litcode $ 130-146 litcode $ 148-155 litcode $ 157-164 litcode $
litcode $ 200-271 htem $ 273-324;
if bodysys="CARDIAC DISORDERS" or aetxt="SUICIDAL" then serious="Y";
else serious="N";
uniqueid="UNI" || put(subject,3.);
datalines;
run;

* adef.sas;
data adef;
set adef;
by usubjid;

retain crit1 "Pain Improvement from baseline of at least 3 points";
RESPFL=put((.xchgcr-2), 1.);
if RESPFL="Y" then crit1fl="Y";
else crit1fl="N";
run;

data adef;
merge adtc.xp(in=inadef) adtl(in=inadtl);
by usubjid;

if not(inadtl and inadef) then put "PROBLEM: Missing subject?-- " usubjid inadef inadtl;

rename trt01prrtp trt01gnrrtpn xptestparam xptestcdparamcd visitvisitn xgorres=aval;
if inadtl and inadef;
avisitn=put(put(visitnum, visitn.), best.);

/* ***** version: macro invocation start of <dtc2dt> ***** dtc2dt(xpdt,
refdt=randtc)*****
*****
/
2
if length(trim(xpdt))=10 and index(xpdt,'--')=0 then adtc=input(xpdt, yymmdd10.);
else if length(xpdt)=16 and index(xpdt,'--')=0 and index(xpdt,':')=0 then do;
adtc=input(trim(xpdt)||"--00", a0001d19.);
adtc=datepart(adtc);
* optionally add formats: ;
format adtc datetime16.;
end;
if (.<adt=randtc then ady=adt - randdt;
else if adt=randtc then ady=adt - randdt+1;
* optionally add formats: ;
format adt yymmdd10.;
/* ***** version: macro invocation start of <dtc2dt> *****
*****
/

```

```

****in this simplistic case all subjects received the treatment they were randomized to;
extttttttt;
run;

* EX.sas;
proc sort data=ex;
by studyid usubjid exttt exttdtc;
run;

* EX.sas;
data ex;
retain STUDYID DOMAIN USUBJID EXSEQ EXTRT EXDOSE EXDOSU EXDOSGRM EPOCH EXSTDTC EXENOTC EXSTOV EXENDV;
set ex(drop=xseq);
by studyid usubjid exttt exttdtc;

if not(first.exstdtc and last.exstdtc) then put "WARN"
"ING": key variables do not define an unique record." usubjid;

retain exseq 0;
exseq=xseq+1;
label exseq="Sequence Number";
run;

* EX.sas;
%make_sort_order(metadatafile=mt/viya-share/homes/tomas-2/demo-study/metadata/chapter02/
SDTM_METADATA.xls, dataset=EX)

* EX.sas;
proc sort data=ex(keep=STUDYID DOMAIN USUBJID EXSEQ EXTRT EXDOSE EXDOSU EXDOSGRM EPOCH EXSTDTC EXENOTC
EXSTOV EXENDV)
out=adtc.ex;
by STUDYID USUBJID EXTRT EXENDTC;
run;

* adtc.sas;
proc sort data=adtc.ex(keep=usubjid exseq) out=latex nodupkey;
by usubjid exseq;

* adtc.sas;
data latex;
set latex;
set latex;
by usubjid exseq;
if last.usubjid;

* adtc.sas;
adtc;
merge adtc(in=inadtc rename=(randtc=rtartdt)) adef(in=inadef) adse(in=inadse) latex(in=inlatex);
by usubjid;
retain param "Time to first pain relief (days)" paramcd "TTPRELFL";
rename trt01prrtp trt01gnrrtpn;

length exseq $10. exdosm $4.;

if (.<chgcr and (adt=astdt or not inadse)) then do;
cnsr="Y";
adtc=adtc;
eventdescript(cnsr, evtmdcc.);
excdow="ADEP";
excvvar="XPDV";
exseq=xpsq;
end;
else if chgcr and (.< adtc=astdt or not inadse) then do;
cnsr="Y";
adtc=adtc;
eventdescript(cnsr, evtmdcc.);
excdow="ADAE";
excvvar="ASTDV";
exseq=xpsq;
end;
else do;
cnsr="Y";
adtc=adtc;
eventdescript(cnsr, evtmdcc.);
excdow="ADGL";
excvvar="TRTEDT";
exseq=xpsq;
end;

aval=adt-startdt+1;
format startdt adt yymmdd10.;
run;

* adtc.sas;
data adtc;
retain STUDYID USUBJID SITEID COUNTRY AGE AGEGR1N AGEGR1 SEX RACE TRTPN TRTP PARAMCD PARAM ADT STARTDT AVAIL
CNSR EVENTDESC SRCDOM SRCVAR SRCSEQ ITTFL;
set EMPTY_adtc adtc;
run;

* adtc.sas;
%make_sort_order(metadatafile=mt/viya-share/homes/tomas-2/demo-study/metadata/chapter06/
ADAM_METADATA.xls, dataset=ADITE)

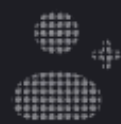
* adtc.sas;
proc sort data=adtc(keep=STUDYID USUBJID SITEID COUNTRY AGE AGEGR1N AGEGR1 SEX RACE TRTPN TRTP PARAMCD
PARAM ADT STARTDT AVAIL CNSR EVENTDESC SRCDOM SRCVAR SRCSEQ ITTFL)
out=adtc.adtc;
by STUDYID USUBJID PARAMCD;
run;

```


Deploy **AI** across the trial analysis lifecycle



Planning



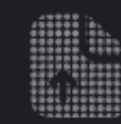
Onboarding



Programming



Validation



Submission



Regulatory review

Imagine what this does to your quality, speed and how many
people you need to achieve it

Available now



Validation automation



SDRG generation



Define-xml generation



Augmented submissions for regulators

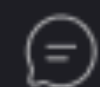
Coming soon



Study planning
automation



Medical writing
automaions



Study
chatbot



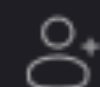
Bug-fixing
agent



Programming
assistant



Automated
analyses



Onboarding



Onboarding