PharmaSUG 2020 - Paper DV-157

Automating of Two Key Components in Analysis Data Reviewer's Guide

Shunbing Zhao, Merck & Co., Inc., Rahway, NJ, USA
Jeff Xia, Merck & Co., Inc., Rahway, NJ, USA

ABSTRACT

The Analysis Data Reviewer's Guide (ADRG) provides reviewers with additional context for analysis datasets received as part of a regulatory submission. It is crucial to submit a clear, concise and precise ADRG. The ADRG consists of seven sections (1. Introduction, 2. Protocol Description, 3. Analysis Considerations Related to Multiple Analysis Datasets, 4. Analysis Data Creation and Processing Issues, 5. Analysis Dataset Descriptions, 6. Data Conformance Summary, 7. Submission of Programs) and optional appendices. In sections 4 and 5, two key components are highly recommended to be included by FDA reviewers: a graph showing data dependencies and a table that identifies efficacy and primary outcome datasets.

This paper presents three useful SAS macros to automatically create two key important components for ADRG. The first macro generates the data dependency graph by using SAS Graph Template Language (GTL). The second macro generates the table of "Analysis Dataset Descriptions" in ADRG. Additionally, the third macro automatically inserts the generated data dependency graph and the table of "Analysis Dataset Descriptions" into the right place in ADRG. This innovative approach removes a few trivial steps from the ADRG generation process, which had to be done manually previously. It helps to create a clear, concise and precise ADRG.

INTRODUCTION

An ADRG is an important part of standards-compliant analysis data submission. The ADRG template, a creation of a PhUSE CSS working group with members from CDISC, pharmaceutical industry and FDA, can be found at www.phusewiki.org/wiki/index.php?title=Analysis_Data_Reviewer%27s_Guide .The ADRG is to assist the reviewers in understanding the submitted data in a usable format. This requires not only comprehensive description of the data, but also a bird's view of the data. In sections 4 and 5 of ADRG template, two key components are highly recommended to be included by FDA reviewers: a graph showing data dependencies and a table that identifies efficacy and primary outcome datasets (Figure 1). Providing visualization of the submitted data through these two components automatically is the focus of this paper.

Hebbar and Matange (2018) presented an approach to create a diagram using SG procedures that underlies the ODS Graphics system. Similarly, the Graph Template Language (GTL) lets you create

dependency graph in a quick and simple manner. In our previous paper (2019), we demonstrate how to process RTF files and manipulate RTF codes using SAS. Here, a table that identifies efficacy and primary outcome datasets will be generated using this method. This paper presents a few useful SAS macros to automatically create two key important components for ADRG, which can be served as a starting point to generate a clear, concise and precise ADRG in a timely and well-controlled way.

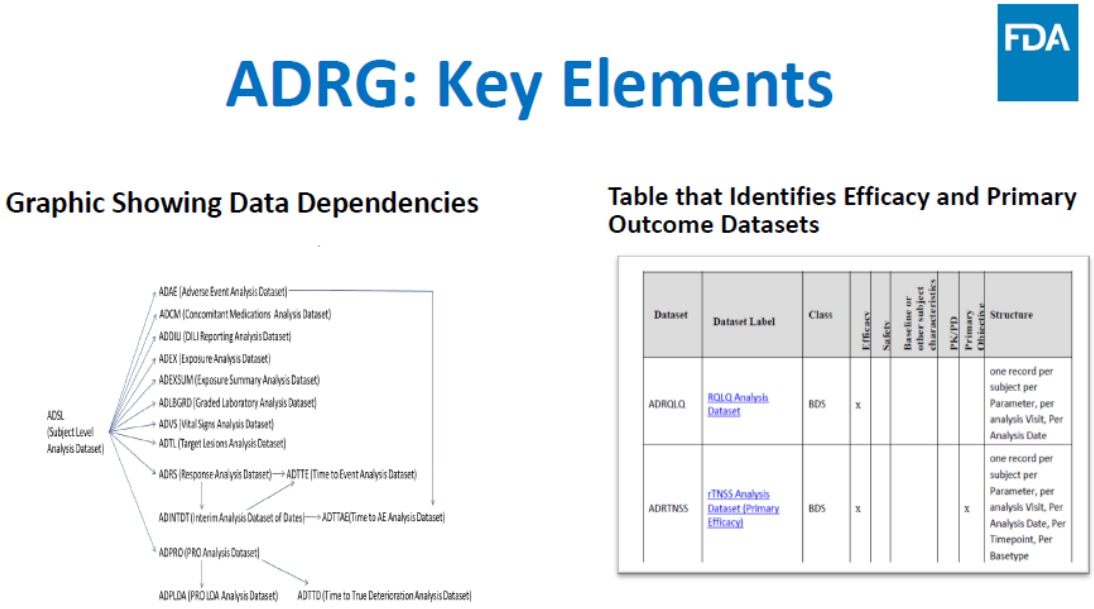


Figure 1. Two key elements recommended by Matilde Kam in FDA Review Process: Recommendations for Review-Ready Submissions to CDER

DISCUSSION

GRAPHIC SHOWING DATA DEPENDENCIES

1. INFORMATION COLLECTION

An Excel spreadsheet is designed to collect information for data dependencies (Table 1). The column 'level' is used to tell the sequential level in data dependencies. In the 'domain' column, datasets in the same level are listed using '|'. The symbol '>' indicates a from-to dependency. Meanwhile, we also import data label information from define.xml using a macro below. We will check consistency between inputs from the Excel spreadsheet and define.xml.

```
%xml2sas(input_xmlfile = fptmt(define.xml)
         ,input_xmlmap = fptmt(adrg.map)
         ,output_libname= work
         ,debug    = Y
         );
```

	A	B	C
1	**level**	**domain**	**label**
2	1	adsl	
3	2	adex>adae	
4	2	adex	
5	2	adcm	
6	2	addili	
7	2	adeff1	
8	2	adeff2	
9	2	adlb	
10	2	adlpd	
11	2	admh	
12	2	adpro	
13	2	advf	
14	3	adlb|adeff1>adresis	

Table 1. Screen print of Excel Spreadsheet used for data dependencies information collection.

2. PLOT GRAPH

After collecting the required information, we need to manipulate the information and construct the data hierarchy for graph plotting. The basic strategy is that coordinates of each text box for each domain are calculated and their dependencies are linked by merging their starting coordinates and end coordinates. We need to prepare a dataset for a text box containing a character variable and two numeric variables: hTextC, xHtc and yHtc that will be used in the TEXTPLOT statement. Another dataset for dependencies linkages of domains containing three numeric variables: linkid, vX and vY is created for the SERIESPLOT statement. Since we have used distinct variables for each plot statement, we can combine two datasets to create the final dependency data set. This combined data set (Table 2) is then used for the GTL plot procedure.

HTEXTC	XHTC	YHTC	LINKID	VX	VY
ADSL^(Subject Level ^ Analysis Dataset)	10	90	210211	43	169.09090909
ADVF^(Viroloqic Failure ^ Analysis Dataset)	43	16.363636364	210211	43	174.54545455
ADPRO^(PRO Analysis Dataset)	43	32.727272727	11211	20	90
ADMH^(Medical History ^ Analysis Dataset)	43	49.090909091	11211	33	180
ADLPD^(Lipids Analysis ^ Dataset)	43	65.454545455	1129	20	90
ADLB^(Safety Lab ^ Analysis Dataset)	43	81.818181818	1129	33	147.27272727
ADEFF2^(CD4 Analysis Dataset)	43	98.181818182	1128	20	90
ADEFF1^(HIV-1 RNA ^ Analysis Dataset)	43	114.54545455	1128	33	130.90909091
ADDILI^(DILI Reporting ^ Analysis Dataset)	43	130.90909091	1127	20	90
ADCM^(Concomitant Medications ^ Analysis Dataset)	43	147.27272727	1127	33	114.54545455
ADEX^(Exposure Analysis ^ Dataset)	43	163.63636364	1126	20	90
ADAE^(Adverse Event ^ Analysis Dataset)	43	180	1126	33	98.181818182
ADRESIS^(Resistance Analysis ^ Dataset)	76	90	11210	20	90
	.	.	11210	33	163.63636364
	.	.	1125	20	90
	.	.	1125	33	81.818181818
	.	.	1124	20	90
	.	.	1124	33	65.454545455
	.	.	1123	20	90
	.	.	1123	33	49.090909091
	.	.	1122	20	90
	.	.	1122	33	32.727272727
	.	.	2731	53	114.54545455
	.	.	2731	66	90
	.	.	2531	53	81.818181818
	.	.	2531	66	90
	.	.	1121	20	90
	.	.	1121	33	16.363636364

Table 2. Screen print of combined dataset of TEXTPLOT and SERIESPLOT.

In the dummy example in this paper, we use the following GTL template.

```
proc template;
 define statgraph dependency;
 begingraph;
    layout overlay/
             xaxisopts=(linearopts=(viewmin=0 viewmax=100) display= none  offsetmax=0
        offsetmin=0)
              yaxisopts=(griddisplay=off linearopts=(viewmin=10 viewmax=200) display=none
        offsetmax=0 offsetmin=0)
              ;
         seriesplot x=vX y=vY / group=linkid arrowheadposition=end arrowheadshape=open
              arrowheadscale=0.75 lineattrs=(thickness=2 color=bigb  pattern=solid);

        textplot x=xHtc y=yHtc text=hTextC / textattrs=(size=7 weight=bold) splitchar='^'
              splitpolicy=splitalways STRIP=TRUE;
      endlayout;
  endgraph;
 end;
run;
```

The graph showing data dependencies from the above template program is generated using PROC SGRENDER as shown in Figure 2 below.

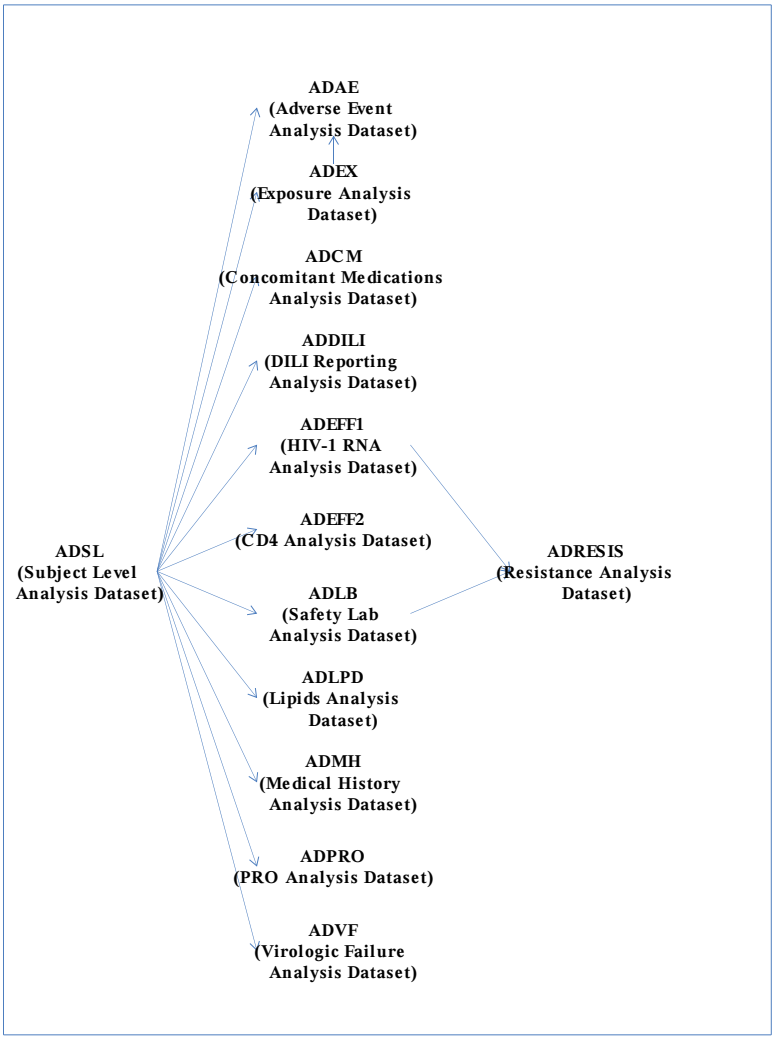


Figure 2. A sample graph showing data dependencies.

After generating the graph, further adjustments may be required. Additionally, a dataset containing programmatically calculated coordinates of dependencies linkages is exported into a new spreadsheet in the same input Excel file. This dataset provides an option for users to tune up positions of linkage arrows in the graph.

3. CONSTRUCT TABLE

Firstly, another Excel spreadsheet is designed to collect information for the table that identifies efficacy and primary outcome datasets (Table 3). The class and structure information of datasets can be also imported from define.xml using the macro %xml2sas. Additionally, we cross-check information between the input Excel spreadsheet and define.xml.

	A	B	C	D	E	F	G	H	I
1	adam_dataset	class	efficacy	safety	baseline	pkpd	primary	structure	hyperlink
2	ADSL	ADSL			X				Y
3	ADAE	OCCDS		X					Y
4	ADCM	OCCDS		X					
5	ADDILI	BDS		X					Y
6	ADEFF1	BDS	X				X		Y
7	ADEFF2	BDS	X						Y
8	ADEX	OTHER		X					Y
9	ADLB	BDS		X					Y
10	ADLPD	BDS		X					Y
11	ADMH	OCCDS		X					
12	ADPRO	BDS	X	X					Y
13	ADVF	BDS	X						Y

Table 3. Screen print of Excel Spreadsheet used for the table that identifies efficacy and primary outcome datasets.

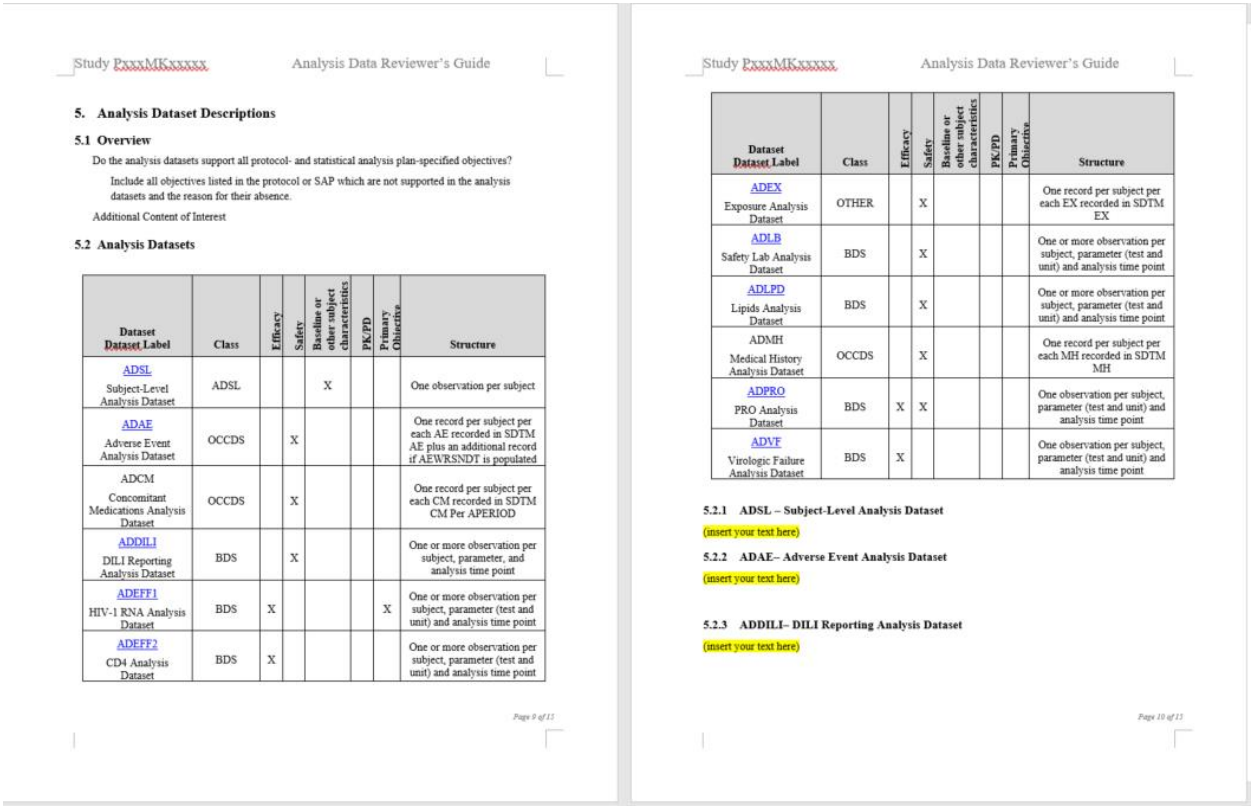
Secondly, macro variables are created at this step to hold all information to display in the table. Below is a piece of code that creates macro variables to hold the label, class, structure and check-box information of other columns in the table.

```
data _null_;
set sect5_22 end=last;
call symput('_adam_ds'||strip(put(seq, best.)), strip(adam_dataset));
call symput('_adam_des'||strip(put(seq, best.)), strip(compbl(label)));
call symput('_adam_class'||strip(put(seq, best.)), strip(class));
call symput('_adam_eff'||strip(put(seq, best.)), strip(efficacy));
call symput('_adam_saf'||strip(put(seq, best.)), strip(safety));
call symput('_adam_base'||strip(put(seq, best.)), strip(baseline));
call symput('_adam_pkpd'||strip(put(seq, best.)), strip(pkpd));
call symput('_adam_prim'||strip(put(seq, best.)), strip(primary));
call symput('_adam_struc'||strip(put(seq, best.)), strip(compbl(structure)));
call symput('_adam_link'||strip(put(seq, best.)), strip(hyperlink));
call symput('_adam_linkseq'||strip(put(seq, best.)), strip(put(hyperseq, best.)));
if last then do;
call symput('_adam_n', strip(put(seq, best.)));
call symput('_adam_hyper_n', strip(put(hyperseq, best.)));
end;
run;
```

Lastly, we can use SAS macro to write RTF codes with resolved macro variables. This means that we can create the designed table row by row through outputting RTF codes and resolved macro variables. Thus, the content can be displayed by a word processer. This paper does not go into details on this topic. Here is an example of partial RTF code which generates a row in the table.

```
…
do rtfcode=
%if %superq(adam_link)=Y %then %do;
"\pard \ltrpar\ql \li0\ri0\widctlpar\intbl\wrapdefault\aspalpha\aspnum\faauto\adjustright\rin0\lin0",
"{\rtlch\fcs1 \af0\afs20 \ltrch\fcs0 \fs20 \trowd
\irow&adam_id\irowband&adam_id\ltrrow\ts11\trgaph115\trrh453\trleft173\trkeep\trbrdrt\brdrs\brdrw10 \trbrdrl",
"\brdrs\brdrw10 \trbrdrb\brdrs\brdrw10 \trbrdrr\brdrs\brdrw10 \trbrdrh\brdrs\brdrw10
\trbrdrv\brdrs\brdrw10\trftsWidth1\trftsWidthB3\trftsWidthA3",
"\trpaddl115\trpaddr115\trpaddfl3\trpaddft3\trpaddfb3\trpaddfr3\tblrsid13719589\tbllkhdrrows\tbllkhdrcols\tblind288\tblindtype3
\clvertalc\clbrdrt\brdrs\brdrw10 \clbrdrl\brdrs\brdrw10 \clbrdrb\brdrs\brdrw10 \clbrdrr",
"\brdrs\brdrw10 \clcbpat8\cltxlrtb\clftsWidth3\clwWidth2167\clcbpatraw8 \cellx2340\clvertalc\clbrdrt\brdrs\brdrw10
\clbrdrl\brdrs\brdrw10 \clbrdrb\brdrs\brdrw10 \clbrdrr\brdrs\brdrw10 \cltxlrtb\clftsWidth3\clwWidth1350\clshdrawnil
\cellx3690\clvertalc",
"\clbrdrt\brdrs\brdrw10 \clbrdrl\brdrs\brdrw10 \clbrdrb\brdrs\brdrw10 \clbrdrr\brdrs\brdrw10
\cltxlrtb\clftsWidth3\clwWidth450\clshdrawnil \cellx4140\clvertalc\clbrdrt\brdrs\brdrw10 \clbrdrl\brdrs\brdrw10
\clbrdrb\brdrs\brdrw10 \clbrdrr\brdrs\brdrw10",
"\cltxlrtb\clftsWidth3\clwWidth450\clshdrawnil \cellx4590\clvertalc\clbrdrt\brdrs\brdrw10 \clbrdrl\brdrs\brdrw10
\clbrdrb\brdrs\brdrw10 \clbrdrr\brdrs\brdrw10 \cltxlrtb\clftsWidth3\clwWidth900\clshdrawnil
\cellx5490\clvertalc\clbrdrt\brdrs\brdrw10 \clbrdrl",
"\brdrs\brdrw10 \clbrdrb\brdrs\brdrw10 \clbrdrr\brdrs\brdrw10 \cltxlrtb\clftsWidth3\clwWidth450\clshdrawnil
\cellx5940\clvertalc\clbrdrt\brdrs\brdrw10 \clbrdrl\brdrs\brdrw10 \clbrdrb\brdrs\brdrw10 \clbrdrr\brdrs\brdrw10",
"\cltxlrtb\clftsWidth3\clwWidth540\clshdrawnil \cellx6480\clvertalc\clbrdrt\brdrs\brdrw10 \clbrdrl\brdrs\brdrw10
\clbrdrb\brdrs\brdrw10 \clbrdrr\brdrs\brdrw10 \cltxlrtb\clftsWidth3\clwWidth2880\clshdrawnil \cellx9360\row \ltrrow}\pard
\ltrpar",
"\qc \li0\ri0\sb120\widctlpar\intbl\wrapdefault\aspalpha\aspnum\faauto\adjustright\rin0\lin0\pararsid1970197",
"{\field\flddirty{\*\fldinst {\rtlch\fcs1 \af0 \ltrch\fcs0 \dbch\af14 \hich\af0\dbch\af14\loch\f0 HYPERLINK  \\l
&double_quote._5.2.&adam_linkseq._&adam_ds._",
"\hich\f0 \endash \loch\f0 &double_quote}{\rtlch\fcs1 \af0 \ltrch\fcs0 \dbch\af14 }}{\fldrslt {\rtlch\fcs1 \af0",
"\ltrch\fcs0 \cs38\ul\cf2\dbch\af14\hich\af0\dbch\af14\loch\f0 &adam_ds\par }}}",
"\pard\plain \ltrpar\qc \li0\ri0\sb120\widctlpar\intbl\wrapdefault\aspalpha\aspnum\faauto\adjustright\rin0\lin0\pararsid1970197
\rtlch\fcs1 \af0\afs22\alang1025 \ltrch\fcs0 \fs22\lang1033\langfe1033\cgrid\langnp1033\langfenp1033 \sectd \ltrsect",
"\psz1\linex0\headery1008\footery1008\endnhere\titlepg\pgbrdropt32\sectlinegrid360\sectdefaultcl\sectrsid9584432\sftnbj
{\rtlch\fcs1 \af0 \ltrch\fcs0 &adam_des\cell &adam_class\cell &adam_eff\cell &adam_saf\cell &adam_base\cell &adam_pkpd\cell
&adam_prim\cell }",
"{\rtlch\fcs1 \af0 \ltrch\fcs0\dbch\af14 \hich\af0\dbch\af14\loch\f0 &&_adam_struc&adam_id\cell }";
%end;
…
```

The output from the above approach is shown in Figure 3 below.



Study PxxxMKxxxxx Analysis Data Reviewer's Guide

5. Analysis Dataset Descriptions

5.1 Overview

Do the analysis datasets support all protocol- and statistical analysis plan-specified objectives?

Include all objectives listed in the protocol or SAP which are not supported in the analysis datasets and the reason for their absence.

Additional Content of Interest

5.2 Analysis Datasets

Dataset Dataset Label	Class	Efficacy	Safety	Baseline or other subject characteristics	PK/PD	Primary Objective	Structure
ADSL Subject-Level Analysis Dataset	ADSL			X			One observation per subject
ADAE Adverse Event Analysis Dataset	OCCDS		X				One record per subject per each AE recorded in SDTM AE plus an additional record if AEWRSNDT is populated
ADCM Concomitant Medications Analysis Dataset	OCCDS		X				One record per subject per each CM recorded in SDTM CM Per APERIOD
ADDILI DILI Reporting Analysis Dataset	BDS		X				One or more observation per subject, parameter, and analysis time point
ADEFF1 HIV-1 RNA Analysis Dataset	BDS	X				X	One or more observation per subject, parameter (test and unit) and analysis time point
ADEFF2 CD4 Analysis Dataset	BDS	X					One or more observation per subject, parameter (test and unit) and analysis time point

Page 9 of 13

Study PxxxMKxxxxx Analysis Data Reviewer's Guide

Dataset Dataset Label	Class	Efficacy	Safety	Baseline or other subject characteristics	PK/PD	Primary Objective	Structure
ADEX Exposure Analysis Dataset	OTHER		X				One record per subject per each EX recorded in SDTM EX
ADLB Safety Lab Analysis Dataset	BDS		X				One or more observation per subject, parameter (test and unit) and analysis time point
ADLPD Lipids Analysis Dataset	BDS		X				One or more observation per subject, parameter (test and unit) and analysis time point
ADMH Medical History Analysis Dataset	OCCDS		X				One record per subject per each MH recorded in SDTM MH
ADPRO PRO Analysis Dataset	BDS	X	X				One observation per subject, parameter (test and unit) and analysis time point
ADVF Virologic Failure Analysis Dataset	BDS	X					One observation per subject, parameter (test and unit) and analysis time point

5.2.1 ADSL – Subject-Level Analysis Dataset

(insert your text here)

5.2.2 ADAE– Adverse Event Analysis Dataset

(insert your text here)

5.2.3 ADDILI– DILI Reporting Analysis Dataset

(insert your text here)

Page 10 of 13

Figure 3. Screen print of a sample table that identifies efficacy and primary outcome datasets.

4. INSERT GRAPH AND TABLE INTO TEMPLATE

At this step, the RTF template of ADRG is now read into SAS using the INFILE and INPUT statements in a DATA step. Meanwhile, the table created in RTF codes is inserted into the template as well. Furthermore, we find the appropriate RTF codes and then replace the dependency graph example with the graph created in step 2. The search and replace operations will be handled partially by the codes below.

```
…
data dependency;
   set dependency;
   retain keepn;
   if index(rtfcode, '\*\shppict{\pict\')>0 then do;
      keepn=1;
      pattern1= prxparse("/\\picwgoal(\d+)/");
      call prxsubstr(pattern1,rtfcode,start1,length1);
      call symput("picwgoal",strip(put(input(compress(substr(rtfcode, start1,
length1),,'kd'),best.),best.)));
      pattern2= prxparse("/\\pichgoal(\d+)/");
      call prxsubstr(pattern2,rtfcode,start2,length2);
      call symput("pichgoal",strip(put(input(compress(substr(rtfcode, start2,
length2),,'kd'),best.),best.)));
    end;
    if index(rtfcode, '}}}')>0 then keepn=.;
    if keepn=1 and index(rtfcode, '\*\shppict{\pict\')<=0 ;
    drop pattern1 start1 length1 pattern2 start2 length2 keepn;
run;
data adrg1 adrg2;
   set adrg;
   retain part1 part2;
   if
index(rtfcode,'\picscalex100\picscaley100\piccropl0\piccropr0\piccropt0\piccropb0\picw15850\pich9
950\picwgoal8986\pichgoal5641\pngblip\bliptag-528849072{\*\blipuid
e07a675087a9031cb8700263a34639cf}') then do;
     part1=_n_;
     rtfcode=tranwrd(rtfcode, "picwgoal8986", "picwgoal&picwgoal");
     rtfcode=tranwrd(rtfcode, "pichgoal5641", "pichgoal&pichgoal");
     rtfcode=tranwrd(rtfcode, "\pngblip", "\emfblip");
   end;
   if index(rtfcode,'}}}}{\rtlch\fcs1 \af0 \ltrch\fcs0 \insrsid471234\charrsid15672794') then do;
      part2=_n_;
      rtfcode="}}}}{\rtlch\fcs1 \af0 \ltrch\fcs0 \insrsid471234\charrsid15672794";
   end;
   if _n_<=part1 or missing(part1) then output adrg1;
   if _n_>=part2>. then output adrg2;
run;
data _null_;
  file "&out_path/&out_adrg..rtf" lrecl=32767 nopad;
  set adrg1 dependency adrg2;
  put  rtfcode ;
run;
```

CONCLUSION

This paper presents a few useful SAS macros to automatically create two key important components for ADRG. Similarly, these approaches can be implemented into other sections in the ADRG. This paper may be just a starting point to generate a clear, concise and precise ADRG in a timely and well-controlled way.

REFERENCES

Matilde Kam, FDA Review Process: Recommendations for Review-Ready Submissions to CDER, *CDISC US Interchange 2019*.

PhUSE Analysis Data Reviewer's Guide Completion Guidelines: https://www.phusewiki.org/docs/Deliverables/ADRG/ADRG_Completion_Guidelines_v1.2%20(4).pdf

RTF 1.9.1 specification, March 2008, from Microsoft.

https://en.wikipedia.org/wiki/Rich_Text_Format

Prashant Hebbar and Sanjay Matange. 2018. "CONSORT Diagrams with SG Procedures". *PharmaSUG conference.*

Lugang Xie (Larry). 2019. "A Simple SAS Utility to Combine Existing RTF Tables/Figures and Create a MultiLevel Bookmark Hierarchy and a Hyperlinked TOC". *PharmaSUG conference.*

Shunbing Zhao, Jeff Xia, and Chao Su. 2019. "Automation of Review Process". *PharmaSUG conference.*