

Data Handling: Paper DH09

Metadata-based Auto-Programming by Python and SAS

Harry Chen
Janssen R&D China
PHUSE US Connect 2021

Agenda



**Metadata-Driven
Approach**



SAS vs. Python



SAS -> Python

Compiler
Checker
Dependency



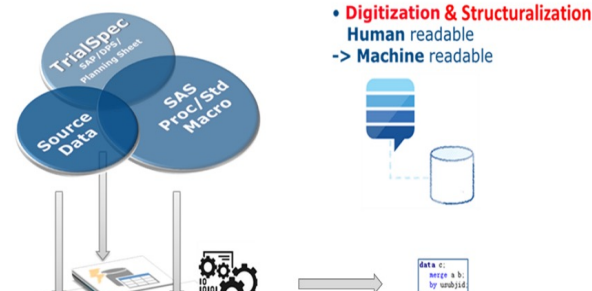
Conclusion

Metadata-Driven Approach

Mapping Specifications

- Excel spreadsheet containing all:
 - Source variables
 - Target DRM variables
 - Mapping specifications
 - Translation of the specifications into SAS (pseudo-)code needed to generate the DRM variables

DATASET	VARIABLE	DRM_DS	DRM_VAR	SPECIFICATION	FUNCTION
source_ae_gi_900yn	PROJECT	AE	STUDYID	Recode according to STUDYID recoding list	RECODE (STUDYID)
source_ae_gi_900yn	STUDYID			Not mapped	NOT MAPPED
source_ae_gi_900yn	SUBJECT	AE	SUBJID	Left justify and uppercase source variable	FUNCTION (SUBJID = STRIP(UPCASE(SUBJECT)));
source_ae_gi_900yn	SITENUMBER	AE	SITEID	Left justify and uppercase source variable	FUNCTION (SITEID = STRIP(UPCASE(SITENUMBER)));
source_ae_gi_900yn	INSTANCENAME	AE	VISIT	Recode according to VISIT recoding list	RECODE (VISIT)
source_ae_gi_900yn		AE	AECAT	Assign value 'ADVERSE EVENTS/SERIOUS AEs'	FUNCTION (AECAT = 'ADVERSE EVENTS/SERIOUS AEs');
source_ae_gi_900yn	AEYN	AE	AEYN	Copy from source variable	COPY
source_ae_gi_900yn	AEYN_STD			Not mapped	NOT MAPPED
source_ae_gi_900	PROJECT	AE	STUDYID	Recode according to STUDYID recoding list	RECODE (STUDYID)
source_ae_gi_900	STUDYID			Not mapped	NOT MAPPED
source_ae_gi_900	SUBJECT	AE	SUBJID	Left justify and uppercase source variable	FUNCTION (SUBJID = STRIP(UPCASE(SUBJECT)));
source_ae_gi_900	SITENUMBER	AE	SITEID	Left justify and uppercase source variable	FUNCTION (SITEID = STRIP(UPCASE(SITENUMBER)));
source_ae_gi_900	INSTANCENAME	AE	VISIT	Recode according to VISIT recoding list	RECODE (VISIT)
source_ae_gi_900	AETERM	AE	AETERM	Copy from source variable	COPY
source_ae_gi_900	AECAT	AE	AESPINT	Copy from source variable	COPY
source_ae_gi_900	AECAT_STD			Not mapped	NOT MAPPED
source_ae_gi_900	AESEV	AE	AESEV	Copy from source variable	COPY
source_ae_gi_900	AEREL	AE	AEREL	Copy from source variable	COPY
		AE		POSTSTEP1: COMBINE SOURCE DATASETS. Combine the mapped source datasets ae_gi_900yn and ae_gi_900 by merging on calculated values of STUDYID, SITEID and SUBJID.	POSTSTEP1: PROC sort DATA=work.mapped_source_ae_gi_900; BY studyid siteid subjid; RUN; PROC sort DATA=work.mapped_source_ae_gi_900yn; BY studyid siteid subjid; RUN; DATA work.mapped_combined_ae1; MERGE work.mapped_source_ae_gi_900 work.mapped_source_ae_gi_900yn; BY studyid siteid subjid; RUN;



<https://www.lexjansen.com/phuse/2018/si/SI15.pdf>

SAS vs. Python



- Common language for data manipulation

One data structure: Table

Two data types: Char/Num

Implicit Loop in data step

	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69	112.5
2	Alice	F	13	56.5	84
3	Barbara	F	13	65.3	98
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5
6	James	M	12	57.3	83
7	Jane	F	12	59.8	84.5
8	Janet	F	15	62.5	112.5
9	Jeffrey	M	13	62.5	84
10	John	M	12	59	99.5
11	Joyce	F	11	51.3	50.5
12	Judy	F	14	64.3	90
13	Louise	F	12	56.3	77
14	Mary	F	15	66.5	112
15	Philip	M	16	72	150
16	Robert	M	12	64.8	128
17	Ronald	M	15	67	133
18	Thomas	M	11	57.5	85
19	William	M	15	66.5	112

Data Tasks

- ▶ List Table Attributes
- ▶ Characterize Data
- ▶ Describe Missing Data
- ▶ List Data
- ▶ Transpose Data
- ▶ Stack/Split Columns
- ▶ Filter Data
- ▶ Select Random Sample
- ▶ Partition Data
- ▶ Sort Data
- ▶ Rank Data
- ▶ Transform Data
- ▶ Standardize Data
- ▶ Recode Values
- ▶ Recode Ranges

Source: SAS® Studio 3.71 with SAS® 9.4 and SAS® Viya® Programming Documentation/ SAS® Studio 3.71: Task Reference Guide

SAS & Python



Use Python for...

>>> More

Web Development: Django , Pyramid , Bottle , Tornado , Flask , web2py

GUI Development: tkinter , PyGObject , PyQt , PySide , Kivy , wxPython

Scientific and Numeric: SciPy , Pandas , IPython

Software Development: Buildbot , Trac , Roundup

System Administration: Ansible , Salt , OpenStack

<https://www.python.org>

Compiler

Code generation

```
/*<FINAL DATASET>*/
DATA mapped_curr1b._currds (KEEP=&manualkeepstr. &tovarkeepstr. &keepstraftermerge. &renaftermerge. &autorekeepstr.

ATTRIB &chars. &numerics.;

LENGTH RH_FROM_NUM RH_TO_NUM 8. RH_CODELIST RH_FROM_CHAR RH_TO_CHAR $MAX_CHAR_VAR_LENGTH..;
IF 0=1 THEN CALL MISSING(RH_FROM_NUM, RH_TO_NUM, RH_CODELIST, RH_FROM_CHAR, RH_TO_CHAR);
IF 0=1 THEN CALL MISSING(&targetkeepcomma.);

SET work._currds(KEEP= &keepstr
WHERE=((&wherestr.))
RENAME=((&renstr. &autorekeepstr.
));

&copystr.;
&assignstr.;
&blankstr.;
&concatstr.;
&recostr.;
&execfunc.;

UN;
degen_stop.();
```

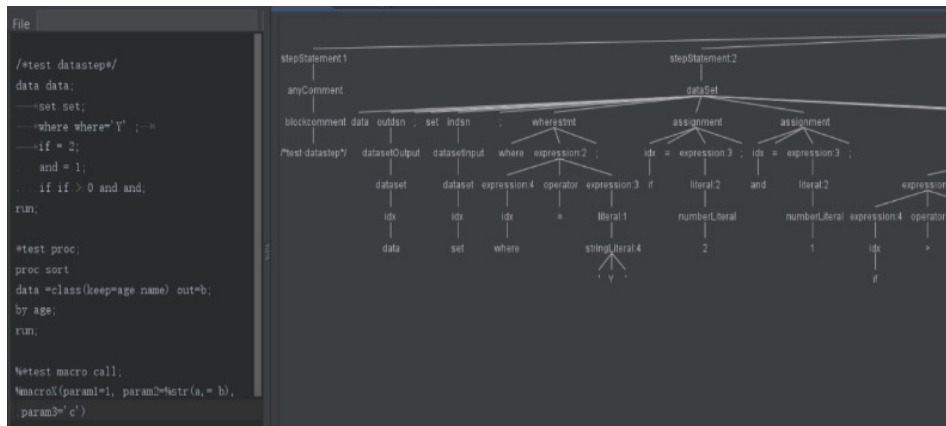
```
_xcode =f"" /*<FINAL DATASET>*/ {nl+''}
DATA work.(strjoin(currds)); {nl+''}
SET {strjoin(currds1b)}; {nl+''}
RUN; {nl+''}
DATA mapped_(strjoin(curr1b))_(strjoin(currds))(KEEP={strjoin(manualkeepstr)} {strjoin(tovarkeepstr)} {strjoin(keepstr)}
{nl+''}
ATTRIB {strjoin(chars)} {strjoin(numerics)}; {nl+''}
{nl+''}
LENGTH RH_FROM_NUM RH_TO_NUM 8. RH_CODELIST RH_FROM_CHAR RH_TO_CHAR ${strjoin(MAX_CHAR_VAR_LENGTH)}; {nl+''}
IF 0=1 THEN CALL MISSING(RH_FROM_NUM, RH_TO_NUM, RH_CODELIST, RH_FROM_CHAR, RH_TO_CHAR); {nl+''}
IF 0=1 THEN CALL MISSING({strjoin(targetkeepcomma)}); {nl+''}
{nl+''}
SET work.{strjoin(currds)}(KEEP= {strjoin(keepstr)} {nl+''}
WHERE=({{strjoin(wherestr)}}) {nl+''}
RENAME=({strjoin(renstr)} {strjoin(autorekeepstr)}) {nl+''}
); {nl+''}
{nl+''}
{strjoin(copystr)}; {nl+''}
{strjoin(assignstr)}; {nl+''}
{strjoin(blankstr)}; {nl+''}
{strjoin(concatstr)}; {nl+''}
{strjoin(recostr)}; {nl+''}
{strjoin(execfunc)}; {nl+''}
{nl+''}
RUN;
```

Checker

Syntax check

Description : Check whether the mapping functions have 'DO' statement
If they have 'END' statements.

```
if first.code then do;  
  do_count = 0;  
  end_count = 0;  
end;  
  
if find(statement, " do;") or find(statement, "%do;") then do_count+1;  
if statement = "end;" or statement = "%end;" then end_count+1;  
  
if last.code and do_count < end_count then output;
```



1_dataset.gif



2_proc.gif



3_macro.gif



4_combined.gif

Dependency

Output data and intermediate data dependency

```
--STEP01
```

```
PROC PROC SQL;
```

```
CREATE TABLE work.mapped_combined_sample3 WORKDS AS
```

```
SELECT a.*, b.rmstdtc FROM work.mapped_combined_samp_Samprtrac WORKDS
```

```
LEFT JOIN outlib.repmiles OUTDS b
```

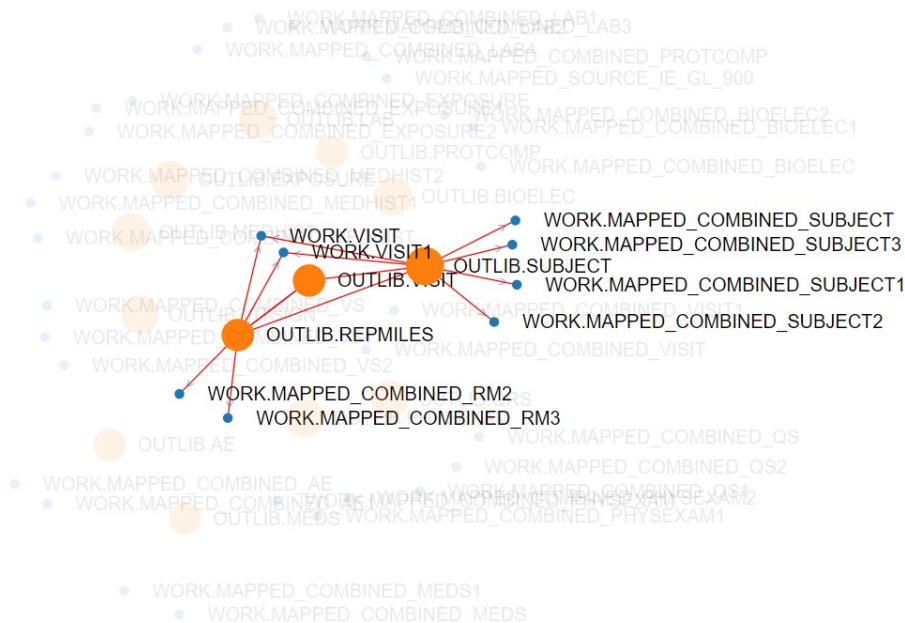
```
ON strip(ucase( a.studyid )) = strip(ucase( b.studyid )) and
```

```
strip(ucase( a.siteid )) = strip(ucase( b.siteid )) and
```

```
strip(ucase( a.subjid )) = strip(ucase( b.subjid )) and
```

```
strip(ucase( b.rmterm )) = 'FIRST STUDY TREATMENT ADMINISTERED';
```

```
QUIT;
```



Conclusion

*Python : manage and translate **metadata** spec into executable code*

*SAS : run the data transformation on the source **data***

Maximize the strength of both language and do a better job together



Thank you

Harry Chen
Janssen R&D China
Phone: 86-21-3337-8892
e-mail: hchen104@its.jnj.com

Maleen Fischer, *Roots*
Maleen, who lives with an incurable and often fatal lung disease called Pulmonary Hypertension, is passionate about utilizing art in order to express herself and her journey with PH.

janssen  | PHARMACEUTICAL COMPANIES OF 