C^2METADATA

Continuous Capture of Metadata

End-to-end process and DDI

Sanda Ionescu
Metadata Specialist
ICPSR

C2Metadata: End-to-end process and DDI

- Project at mid-point: progress overview
  - Tools
  - End-to-end process
  - Updated DDI

C2Metadata: End-to-end process and DDI

Tools developed in the first phase of the project:

- Standard Data Transformation Language – SDTL (Colectica and NSD)
- SPSS parser – SPSS scripts to SDTL (Colectica)
- Stata parser – Stata scripts to SDTL (NSD)
- DDI updater – DDI + SDTL -> revised DDI (MTNA)
- Pseudocode generator – SDTL to human readable text (ICPSR)
- Codebook generator – (revised) DDI to HTML codebook (ICPSR)

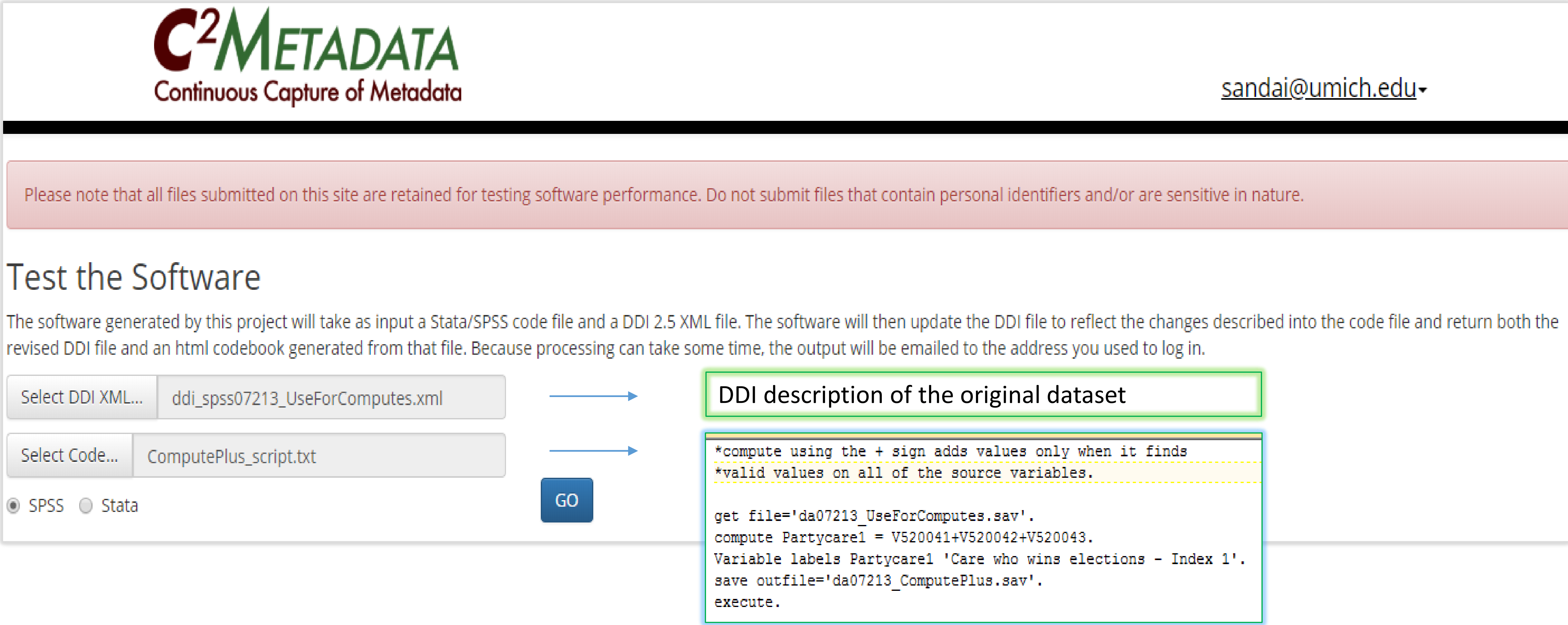
C2Metadata: End-to-end process and DDI

Tools developed in the first phase of the project:

- User-oriented service (ICPSR)
  - Accepts input files (data transformation script and DDI metadata)
  - Accesses the individual tools and runs the end-to-end process
  - Translates the SDTL into human-readable “pseudocode”
  - Generates HTML codebook based on the revised DDI
  - Delivers the output (updated DDI and human-readable codebook) to user

C2Metadata: End-to-end process and DDI

End user service at ICPSR (currently in open Beta testing):



C²METADATA
Continuous Capture of Metadata

Please note that all files submitted on this site are retained for testing software performance. Do not submit files that contain personal identifiers and/or are sensitive in nature.

Test the Software

The software generated by this project will take as input a Stata/SPSS code file and a DDI 2.5 XML file. The software will then update the DDI file to reflect the changes described into the code file and return both the revised DDI file and an html codebook generated from that file. Because processing can take some time, the output will be emailed to the address you used to log in.

Select DDI XML... ddi_spss07213_UseForComputes.xml

Select Code... ComputePlus_script.txt

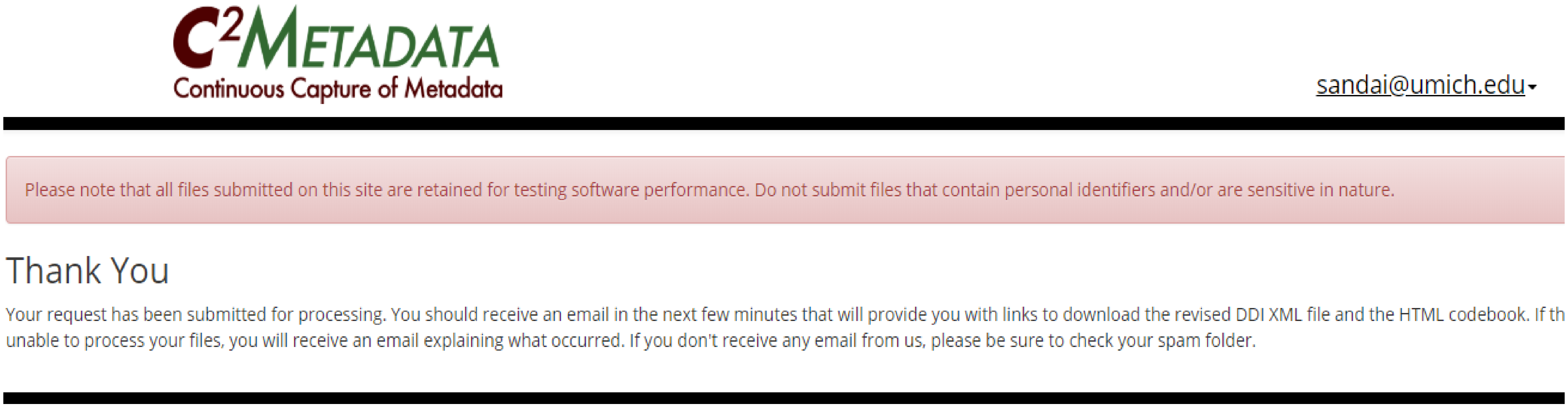
● SPSS ○ Stata

DDI description of the original dataset

```
*compute using the + sign adds values only when it finds
*valid values on all of the source variables.

get file='da07213_UseForComputes.sav'.
compute Partycare1 = V520041+V520042+V520043.
Variable labels Partycare1 'Care who wins elections - Index 1'.
save outfile='da07213_ComputePlus.sav'.
execute.
```

C2Metadata: end user service at ICPSR



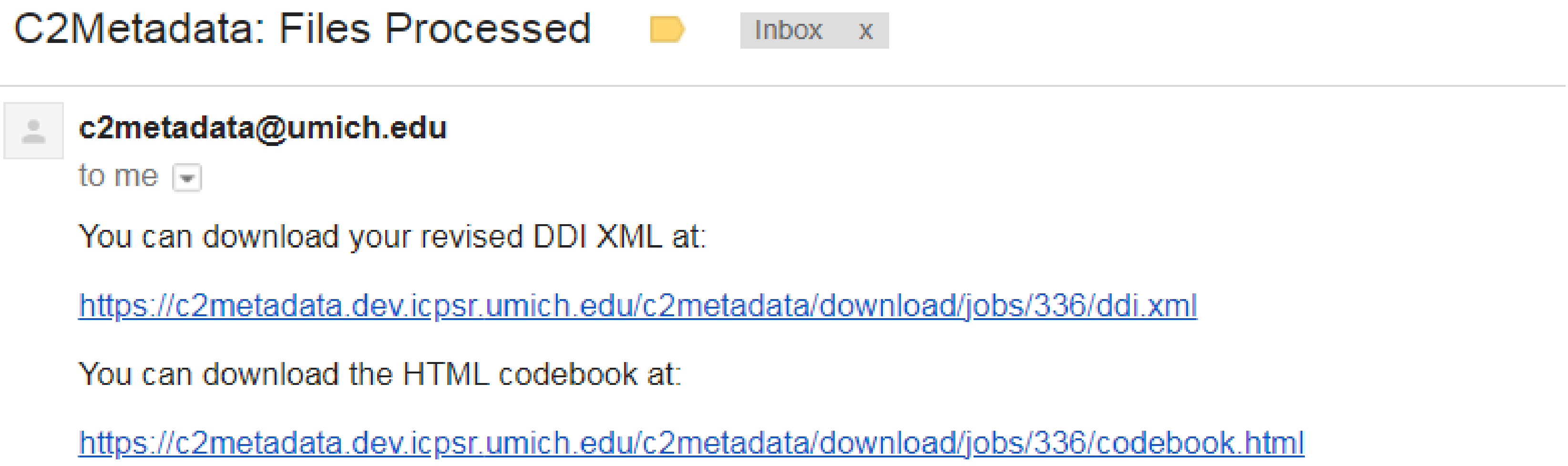
C2METADATA
Continuous Capture of Metadata

sandai@umich.edu

Please note that all files submitted on this site are retained for testing software performance. Do not submit files that contain personal identifiers and/or are sensitive in nature.

Thank You

Your request has been submitted for processing. You should receive an email in the next few minutes that will provide you with links to download the revised DDI XML file and the HTML codebook. If th unable to process your files, you will receive an email explaining what occurred. If you don't receive any email from us, please be sure to check your spam folder.



C2Metadata: Files Processed   Inbox x

c2metadata@umich.edu
to me

You can download your revised DDI XML at:

https://c2metadata.dev.icpsr.umich.edu/c2metadata/download/jobs/336/ddi.xml

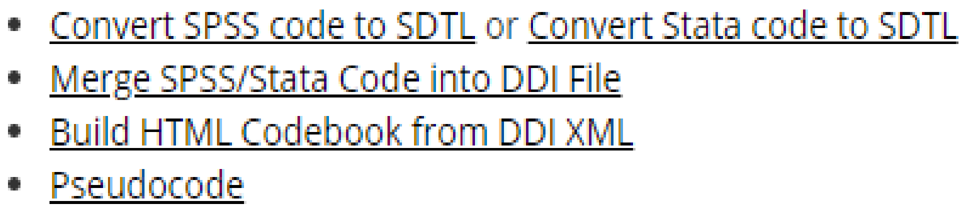
You can download the HTML codebook at:

https://c2metadata.dev.icpsr.umich.edu/c2metadata/download/jobs/336/codebook.html

C2Metadata: end user service at ICPSR

Also available at https://c2metdata.dev.icpsr.umich.edu/c2metadata

- DDI documentation for substantive subsets of
  - The American National Election Study (ANES) cumulative file (1948-2012)
  - The General Social Surveys (GSS) cumulative file (1972-2016)
- For internal testing:
  - Links for testing or working with the individual tools



- Convert SPSS code to SDTL or Convert Stata code to SDTL
- Merge SPSS/Stata Code into DDI File
- Build HTML Codebook from DDI XML
- Pseudocode

  - Links to resources for developers
  - Sample SPSS and Stata scripts for the supported commands
  - Sample DDI input metadata for testing the scripts

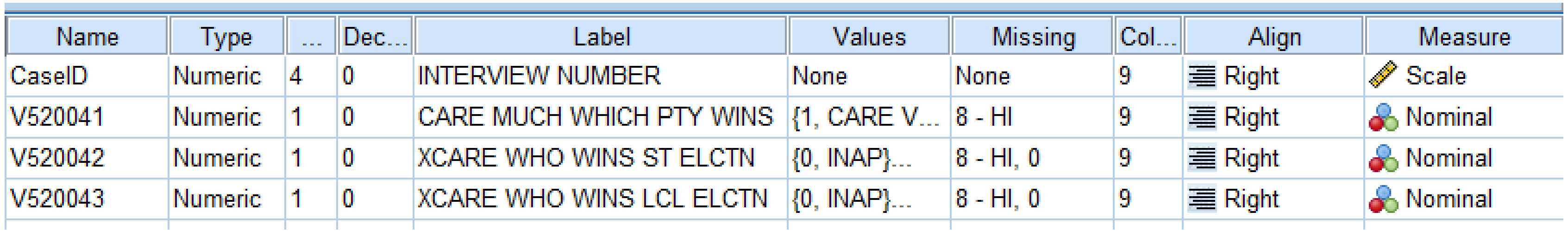
C2Metadata: End-to-end process and DDI

Currently supported commands:

- Recode variable(s)
- Rename variable(s)
- Aggregate variables
- Compute variable(s)
- Delete variable(s)
- Do Repeat Loop
- If – Then Condition
- Add variable label
- Add value labels
- Declare missing values
- Add files
- Match files
- Select cases
- Sort cases
- Sort variables

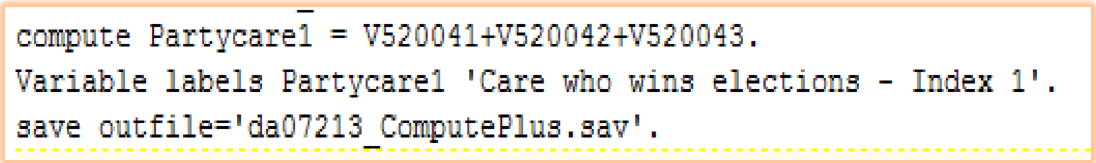
Documenting data transformations in DDI

- Original dataset:



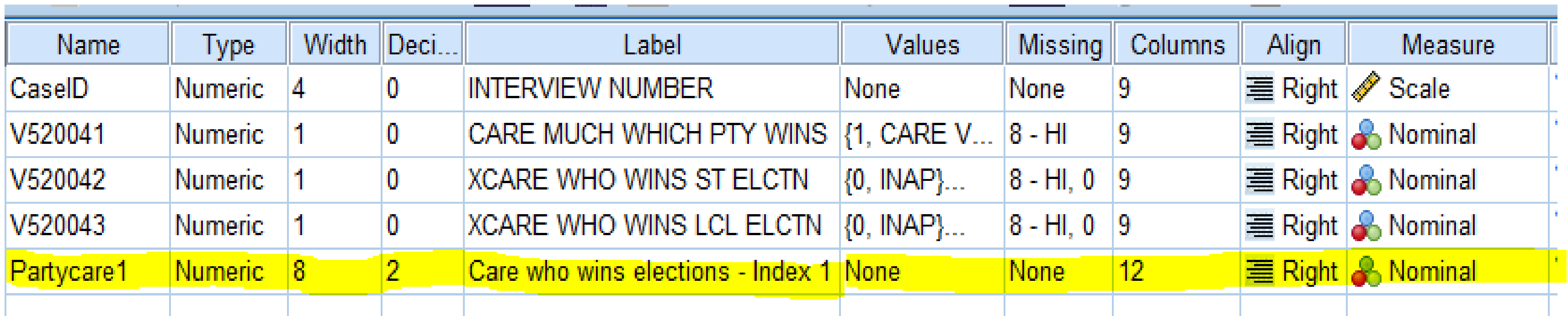
Name	Type	...	Dec...	Label	Values	Missing	Col...	Align	Measure
CaseID	Numeric	4	0	INTERVIEW NUMBER	None	None	9	Right	Scale
V520041	Numeric	1	0	CARE MUCH WHICH PTY WINS	{1, CARE V...	8 - HI	9	Right	Nominal
V520042	Numeric	1	0	XCARE WHO WINS ST ELCTN	{0, INAP}...	8 - HI, 0	9	Right	Nominal
V520043	Numeric	1	0	XCARE WHO WINS LCL ELCTN	{0, INAP}...	8 - HI, 0	9	Right	Nominal

- SPSS script:



```
compute Partycare1 = V520041+V520042+V520043.
Variable labels Partycare1 'Care who wins elections - Index 1'.
save outfile='da07213_ComputePlus.sav'.
```

- Revised dataset:

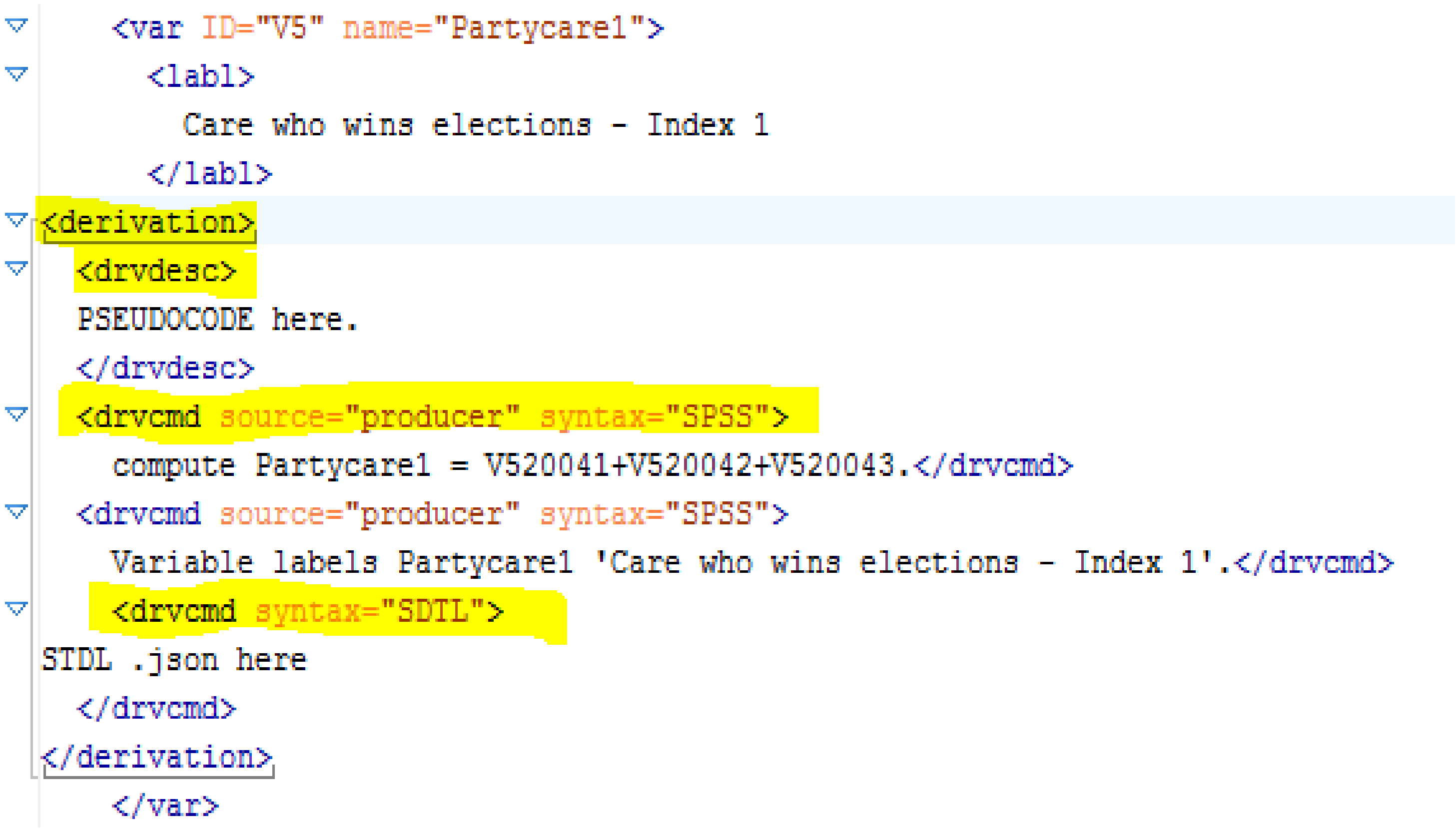


Name	Type	Width	Deci...	Label	Values	Missing	Columns	Align	Measure
CaseID	Numeric	4	0	INTERVIEW NUMBER	None	None	9	Right	Scale
V520041	Numeric	1	0	CARE MUCH WHICH PTY WINS	{1, CARE V...	8 - HI	9	Right	Nominal
V520042	Numeric	1	0	XCARE WHO WINS ST ELCTN	{0, INAP}...	8 - HI, 0	9	Right	Nominal
V520043	Numeric	1	0	XCARE WHO WINS LCL ELCTN	{0, INAP}...	8 - HI, 0	9	Right	Nominal
Partycare1	Numeric	8	2	Care who wins elections - Index 1	None	None	12	Right	Nominal

Documenting data transformations in DDI

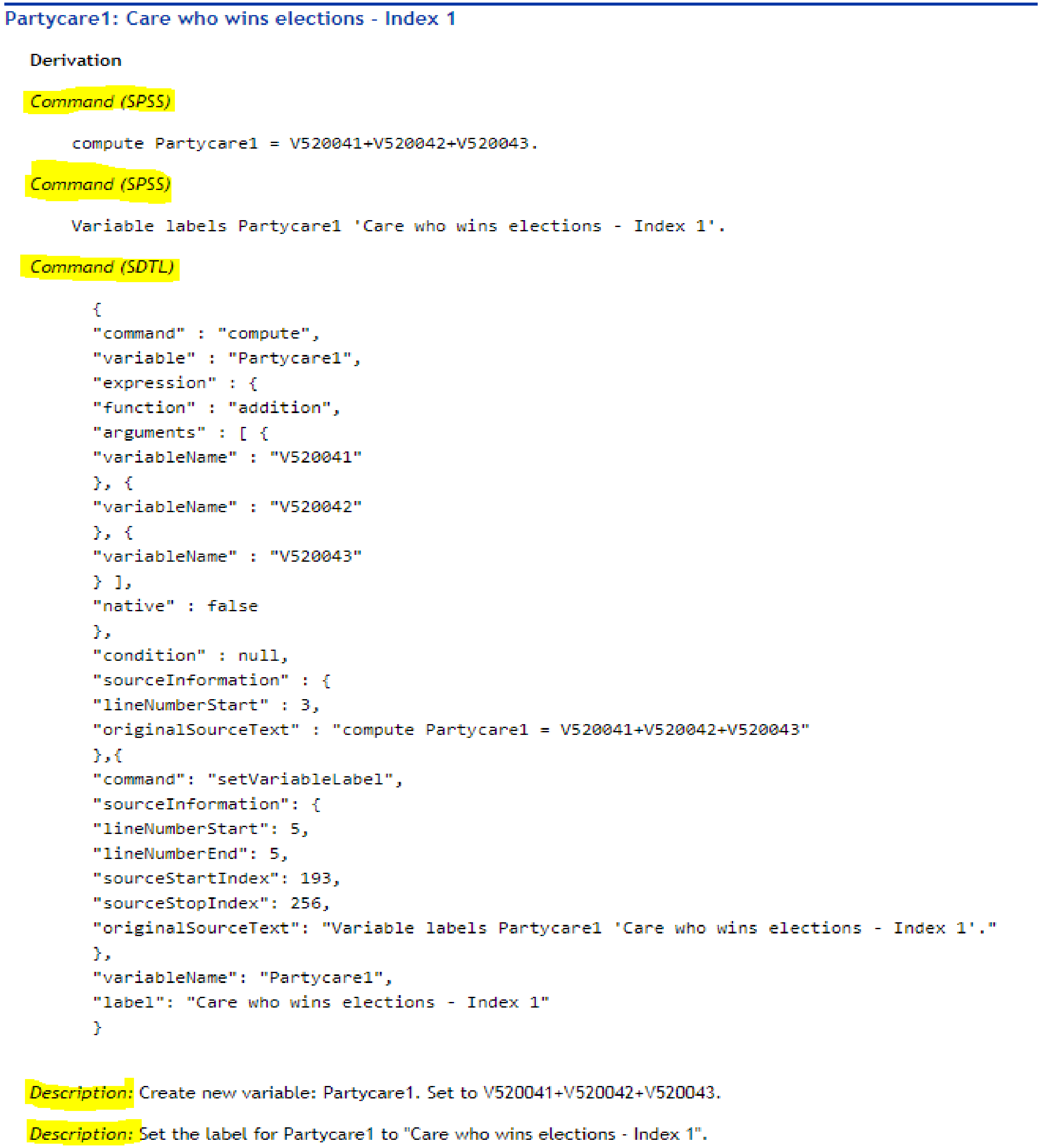
Updated DDI:

- Variable level transformations – use derivation field:



```
    <var ID="V5" name="Partycare1">
      <labl>
        Care who wins elections - Index 1
      </labl>
<derivation>
  <drvdesc>
  PSEUDOCODE here.
  </drvdesc>
  <drvcmd source="producer" syntax="SPSS">
    compute Partycare1 = V520041+V520042+V520043.</drvcmd>
  <drvcmd source="producer" syntax="SPSS">
    Variable labels Partycare1 'Care who wins elections - Index 1'.</drvcmd>
    <drvcmd syntax="SDTL">
STDL .json here
  </drvcmd>
</derivation>
    </var>
```

Documenting data transformations in DDI



Partycare1: Care who wins elections - Index 1

Derivation

Command (SPSS)

```
compute Partycare1 = V520041+V520042+V520043.
```

Command (SPSS)

```
Variable labels Partycare1 'Care who wins elections - Index 1'.
```

Command (SDTL)

```
{
"command" : "compute",
"variable" : "Partycare1",
"expression" : {
"function" : "addition",
"arguments" : [ {
"variableName" : "V520041"
}, {
"variableName" : "V520042"
}, {
"variableName" : "V520043"
} ],
"native" : false
},
"condition" : null,
"sourceInformation" : {
"lineNumberStart" : 3,
"originalSourceText" : "compute Partycare1 = V520041+V520042+V520043"
},{
"command": "setVariableLabel",
"sourceInformation": {
"lineNumberStart": 5,
"lineNumberEnd": 5,
"sourceStartIndex": 193,
"sourceStopIndex": 256,
"originalSourceText": "Variable labels Partycare1 'Care who wins elections - Index 1'."
},
"variableName": "Partycare1",
"label": "Care who wins elections - Index 1"
}
```

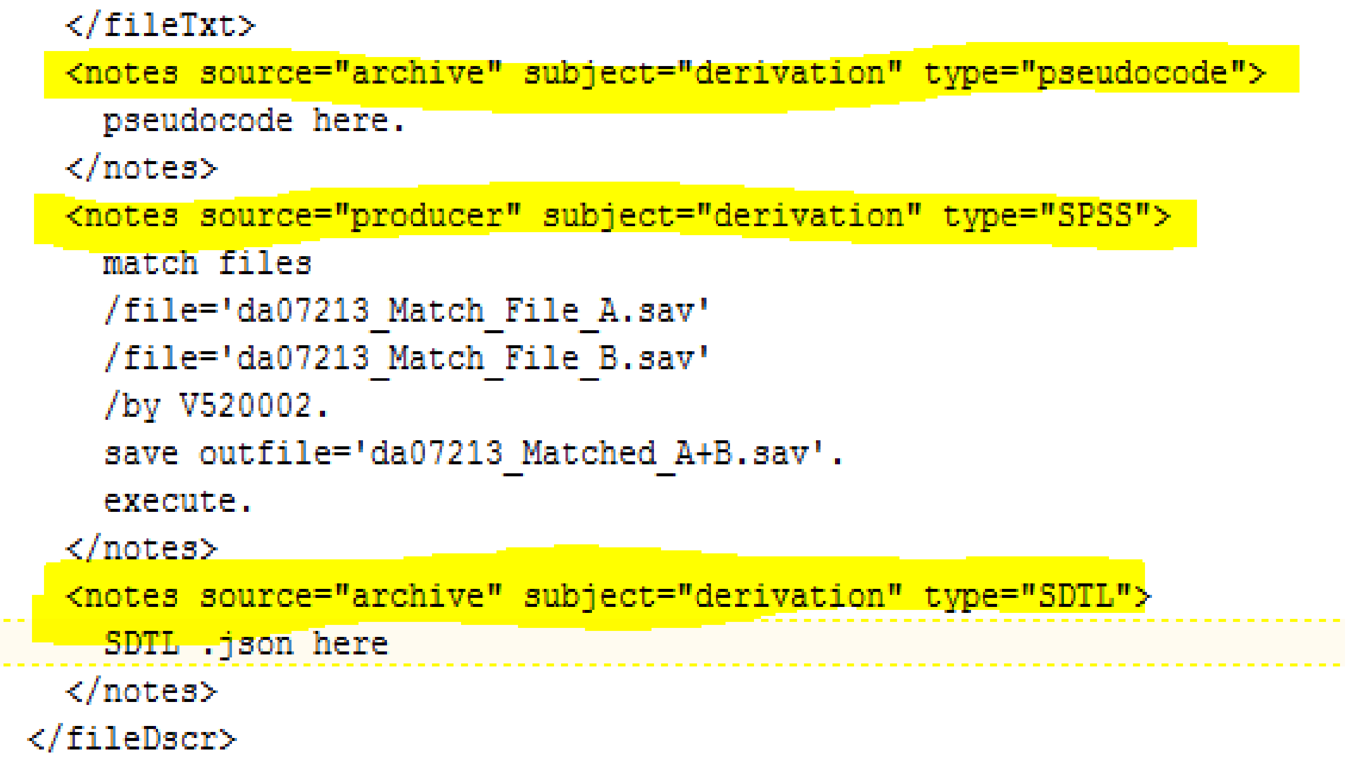
Description: Create new variable: Partycare1. Set to V520041+V520042+V520043.

Description: Set the label for Partycare1 to "Care who wins elections - Index 1".

Documenting data transformations in DDI

Updated DDI:

- File-level transformations: use File Notes with customized content in attributes



```
  </fileTxt>
  <notes source="archive" subject="derivation" type="pseudocode">
    pseudocode here.
  </notes>
  <notes source="producer" subject="derivation" type="SPSS">
    match files
    /file='da07213_Match_File_A.sav'
    /file='da07213_Match_File_B.sav'
    /by V520002.
    save outfile='da07213_Matched_A+B.sav'.
    execute.
  </notes>
  <notes source="archive" subject="derivation" type="SDTL">
    SDTL .json here
  </notes>
</fileDscr>
```

C2Metadata: End-to-end process and DDI

Thank you !

sandai@umich.edu