



PRAHEALTHSCIENCES

**PRA Health
Sciences**

**Completing the
Data set Jigsaw:
Hints and Tips for
Assembling Data
sets Together**

DATE
09-Oct-2017

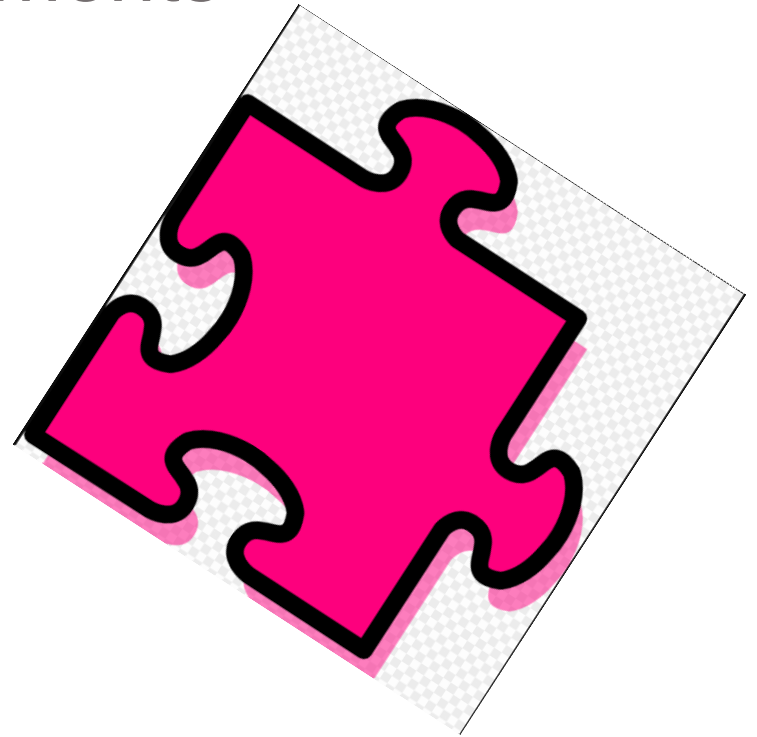
PRESENTED BY
Cerys Rand



What will this presentation cover?

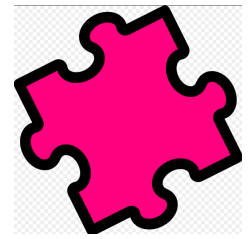
The essential pieces to complete the jigsaw:

- Study Design and Documents
- Source Data Sets
- SAS® Functions
- Summary





Study Design and Documents



Protocol

Blank CRF

Annotated
CRF

DTS

Edit Check
Plan

SDTM
Specifications

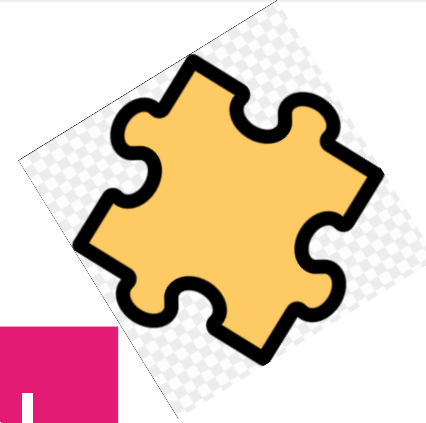
SAP & Shells

ADaM
Specifications





Source Data



CRF
Database

External
Vendor

SDTM

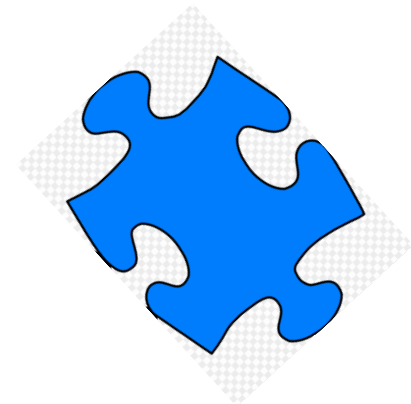
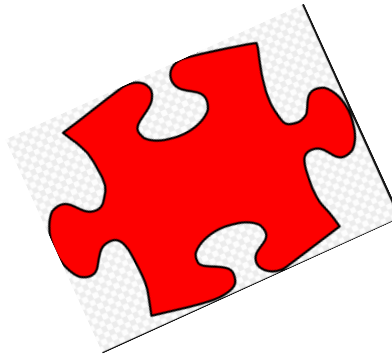
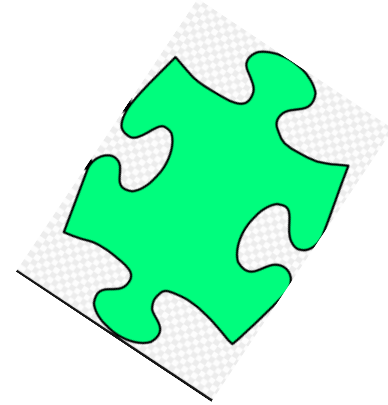
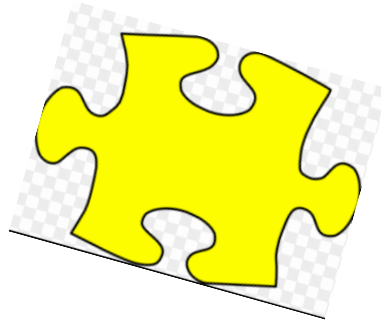
ADaM





SAS Functions

- SET
- MERGE
- UPDATE
- MODIFY





SET

The SET statement can be used in a number of different ways for example, on its own to concatenate data sets or with a BY statement to merge data sets.

Example 1,

Input data sets:

Data set A:

	 subject	 var_a
1	001	1
2	002	2
3	003	3

Data set B:

	 subject	 var_b
1	001	1
2	002	2
3	003	3

Code:

```
data ab;  
  set a b;  
run;
```

Resulting data set:

Data set AB:

	 subject	 var_a	 var_b
1	001	1	.
2	002	2	.
3	003	3	.
4	001	.	1
5	002	.	2
6	003	.	3



SET

Example 2,

Input data sets:

Data set A2:

	 subject	 var_a	 var_b
1	001	1	1
2	003	.	3

Data set B2:

	 subject	 var_a
1	001	1
2	002	2
3	005	3

Code:

```
data ab2;  
  set a2  
  set b2;  
run;
```

Resulting data set:

Data set AB2:

	 subject	 var_a	 var_b
1	001	1	1
2	002	2	3



SET

Example 3,

Input data sets:

Data set A3:

	 subject	 var_a
1	001	1
2	002	2
3	002	3
4	005	3

Data set B3:

	 subject	 var_b
1	001	1
2	002	2
3	003	1
4	003	3

Code:

```
data ab3;  
  set a3 b3;  
  by SUBJECT;  
run;
```

Resulting data set:

Data set AB3:

	 subject	 var_a	 var_b
1	001	1	.
2	001	.	1
3	002	2	.
4	002	3	.
5	002	.	2
6	003	.	1
7	003	.	3
8	005	3	.



MERGE

The MERGE statement can be used to combine multiple data sets together to form one data set in a variety of ways. When using the MERGE statement it is important to ensure that the input data sets are sorted in the order of the variables stated in the BY statement in order to avoid an ERROR message.

Example 4,

Input data sets:

Code:

Resulting data set:

Data set MA4:

	subject	var_a
1	001	1
2	002	2
3	003	3
4	004	4

Data set MB4:

	subject	var_b
1	001	1
2	002	2
3	003	3
4	004	1
5	005	3

```
data mab4;  
  merge ma4 mb4;  
  by SUBJECT;  
run;
```

Data set MAB4:

	subject	var_a	var_b
1	001	1	1
2	002	2	2
3	003	3	3
4	004	4	1
5	005	.	3



MERGE

Example 5,

Input data sets:

Data set MA5:

	 subject	 var_a
1	001	1
2	002	2
3	003	3
4	004	4

Code:

```
data mab5;  
  merge ma5 mb5;  
  by SUBJECT;  
run;
```

Resulting data set:

Data set MAB5:

	 subject	 var_a	 var_b
1	001	1	1
2	002	1	2
3	003	1	3
4	004	1	1
5	005	1	3

Data set MB5:

	 subject	 var_a	 var_b
1	001	1	1
2	002	1	2
3	003	1	3
4	004	1	1
5	005	1	3



MERGE

Example 6,

Input data sets:

Data set MA6:

	 subject	 var_a
1	001	1
2	002	2
3	003	3
4	004	4

Data set MB6:

	 subject	 var_a	 var_b
1	001	1	1
2	002	1	2
3	003	1	3
4	004	1	1
5	005	1	3

Code:

```
data mab6;  
  merge ma6  
        mb6 (rename=(VAR_A=VAR_B1);  
  by SUBJECT;  
run;
```

Resulting data set:

Data set MAB6:

	 subject	 var_a	 var_b1	 var_b
1	001	1	1	1
2	002	2	1	2
3	003	3	1	3
4	004	4	1	1
5	005	.	1	3



MERGE Vs SET

Example 7,

Input data sets:

Data set MA7:

	 subject	 var_a
1	001	1
2	002	2
3	002	3
4	003	3

Data set MB7:

	 subject	 var_a
1	001	1
2	002	2
3	003	1
4	003	3

Code:

```
data MERGE;  
  merge ma7 mb7;  
  by SUBJECT;  
run;
```

Resulting data set:

Data set MERGE:

	 subject	 var_a
1	001	1
2	002	2
3	002	3
4	003	1
5	003	3

Data set SET:

	 subject	 var_a
1	001	1
2	001	1
3	002	2
4	002	3
5	002	2
6	003	3
7	003	1
8	003	3

```
data SET;  
  set ma7 mb7;  
  by SUBJECT;  
run;
```



UPDATE

- Can only be used with a maximum of two input data sets.
- Can be used to replace values from one data set, often known as the master input data set, with values from another secondary input data set, usually known as the transaction data set.
- SAS® requires there to be a BY statement used directly after the UPDATE statement
- It is vital that both input data sets are sorted by the variables in the BY statement.
- Within the master input data set there must be an unique value of the variables in the BY statement, or else only the first value will be updated.



UPDATE

Example 8,

Input data sets:

Data set UA8 (MASTER):

	name	city	country	subject
1	Ericson, Jane	Seattle	USA	0001
2	Dix, Martin	London	UK	0002
3	Gabrielli, Theresa	Paris	FRANCE	0003
4	Clayton, Aria	San Francisco	USA	0004
5	Archuleta, Ruby	London	UK	0005

Code:

```
data uab81;  
  update ua8 ub8;  
  by SUBJECT NAME CITY;  
run;
```

Data set UB8 (TRANSACTION):

	name	city	country	subject
1	Ericson, Jane	Seattle	USA	0001
2	Dix, Martin			0002
3	Gabrielli, Theresa	Paris	FRANCE	0003
4	Clayton, Aria	San Francisco	USA	0004
5	Archuleta, Ruby	London	UK	0005

Resulting data set:

Data set UAB81:

	name	city	country	subject
1	Ericson, Jane	Seattle	USA	0001
2	Dix, Martin			0002
3	Dix, Martin	London	UK	0002
4	Gabrielli, Theresa	Paris	FRANCE	0003
5	Clayton, Aria	San Francisco	USA	0004
6	Archuleta, Ruby	London	UK	0005



UPDATE Vs MERGE

Example 8 - UPDATE

Obs	name	city	country	subject	siteid
1	Ericson, Jane	Seattle	USA	0001	1
2	Dix, Martin	London	UK	0002	1
3	Gabrielli, Theresa	Paris	FRANCE	0003	2
4	Clayton, Aria	San Francisco	USA	0004	2
5	Archuleta, Ruby	London	UK	0005	3

Page Break

Example 8 - MERGE

Obs	name	city	country	subject	siteid
1	Ericson, Jane	Seattle	USA	0001	1
2	Dix, Martin			0002	1
3	Gabrielli, Theresa	Paris	FRANCE	0003	2
4	Clayton, Aria	San Francisco	USA	0004	2
5	Archuleta, Ruby	London	UK	0005	3

```
data update;  
  update ua8 ub8;  
  by SUBJECT NAME;  
run;
```

```
data merge;  
  MERGE ua8 ub8;  
  by SUBJECT NAME;  
run;
```



MODIFY

The MODIFY statement updates observations of the existing data set but the consequence of this is that no updates can be carried out to the structure of that data set. The main purpose of the MODIFY statement is to replace all or part of an observation from one data set with an observation from another.

The MODIFY statement, unlike the MERGE and UPDATE does not require the input data sets to be sorted by the BY statement variables beforehand. The MODIFY statement also allows for there to be duplicate values of the BY variables in both the master and transaction data set unlike the UPDATE statement.

Example 9,

Input data sets:

Code:

Resulting data set:

Data set MOA9:

	subject	var_a	var_b
1	001	1	1
2	002	2	2
3	003	3	3
4	004	4	4

```
data moa9;  
  modify moa9 mob9;  
  by SUBJECT;  
run;
```

Data set MOB9:

	subject	var_a	var_b
1	002	1	5
2	003	1	6
3	004	1	7

Data set MOA9:

	subject	var_a	var_b
1	001	1	1
2	002	1	5
3	003	1	6
4	004	1	7

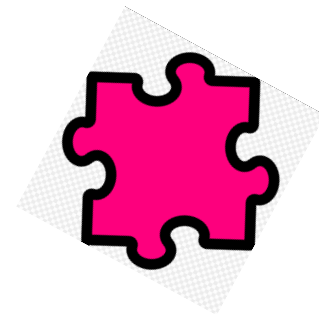
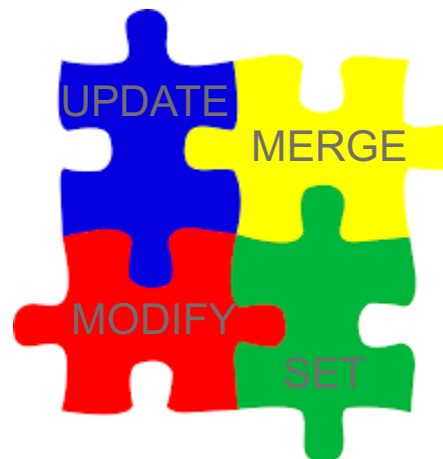


Summary



As can be seen in the examples above often careful consideration will need to be given in order to not overwrite data and therefore data become lost in the combination process. Although the UPDATE and MODIFY statements have their advantages, they can only be used to combine two data sets which when combining numerous raw data sets, are not necessarily the best statement of choice. Neither the MERGE nor the SET statement are not restricted by the number of data sets they can combine.

Each programmer will develop their own preferred way in which to complete the jigsaw that is to assemble data and in several circumstances there is no right or wrong method.





Questions?