MAT7500
Statistical Programming
Fall 2017

Michael A. Posner

SAS Lecture 1

Goals for Today

- **Introduction to Programming and SAS**
- **What you should know already**
- **Reading data**
- **Attributes of data**
- **Combining data**

Introduction to Programming

- Seven recommendations for graduate programs in statistics (from ASA)
  - #1: Solid foundation in methods & theory
  - #2: Programming skills are critical
  - Communication, Teamwork, Non-routine problems, Immersive work experience, Periodic updates
- Data science
  - Most popularly searched for job title (didn't exist 5 years ago)
  - Data management, analysis, subject matter expertise
  - "Jake Porway" is an example of the perennial data scientist – datacon.org. Has BS in Comp Sci, MS/PhD in Stat
  - "Data scientist is a sexed-up term for statistician." Nate Silver
- Job opportunities!

Introduction to SAS

- Started in 1970s
- Used extensively at academic and business environments
- Many analyses and publications use SAS
- SAS invests extensive resources to R & D
- Top 5 companies to work for over many years
- Past-President of ASA is SAS Director of R&D
- National/Regional SUGs

Using SAS at Villanova

- Accessing via Citrixweb
  - The N:/ drive (see handout online)
  - Backing out to your documents (computer, c: drive, user, …)
- Getting SAS installed on your machine
- SAS University?

What You Should Know Already

- Elements of a SAS program
  - "Steps" - DATA, PROCs, (Options)
  - "Statements" within the steps
    - Begin with identifying keyword, end with ;
- SAS Windows
  - Editor, Log, Output, Reviewer
  - To clear reviewer: ods html close; ods html;
- Commenting
  - Using *; or /* and */
- Structure of data tables (rows/columns)

What You Should Know Already

- Character ($) vs. numeric variables
- Missing values
  - "." for numeric, "" or " " for character
- Formats and labels
  - Proc format and format statement, %include
- Titles and footnotes
- Some output control options
  - ps, ls, pageno=1, nodate
- Executing code
  - Submitting and the use of run; (vs. next "step")

What You Should Know Already

- Read through the LOG file!
  - Errors – fatal in that program will abort
  - Warnings – messages that are usually important
  - Notes – messages that may or may not be important
- Getting help with SAS
  - Google "proc"
  - SAS Support – overview, syntax, and examples
    - Google PROC MEANS

SAS Programming 1 Quizzes/Questions

Portions were reproduced with permission of SAS Institutes, Inc, Cary, NC, USA. SAS Institute, Inc. makes no warranties with respect to these materials and disclaims all liability therefor.

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies. Copyright © 2009 SAS Institute Inc. Cary, NC, USA. All rights reserved.

2.01 Quiz

How many steps are in this program?

```
data work.NewSalesEmps;
   length First_Name $ 12
          Last_Name $ 18 Job_Title $ 25;
   infile 'newemps.csv' dlm=',';
   input First_Name $ Last_Name $
         Job_Title $ Salary;
run;

proc print data=work.NewSalesEmps;
run;

proc means data=work.NewSalesEmps;
   class Job_Title;
   var Salary;
run;
```


2.02 Quiz

How does SAS detect the end of the PROC MEANS step?

```
data work.NewSalesEmps;
   length First_Name $ 12
          Last_Name $ 18 Job_Title $ 25;
   infile 'newemps.csv' dlm=',';
   input First_Name $ Last_Name $
         Job_Title $ Salary;
run;

proc print data=work.NewSalesEmps;

proc means data=work.NewSalesEmps;
   class Job_Title;
   var Salary;
```


3.01 Quiz

How many statements are in the DATA step?

a. 1
b. 3
c. 5
d. 7

```
data work.NewSalesEmps;
  length First_Name $ 12
    Last_Name $ 18 Job_Title $ 25;
infile 'newemps.csv' dlm=',';
input First_Name $ Last_Name $
        Job_Title $ Salary;
  run;
```

3.02 Multiple Choice Poll

Which statement is true concerning the DATALINES statement?

a. The DATALINES statement is used when reading data located in a raw (external) data file.
b. The DATALINES statement is used when reading data located directly in the program.

16

3.03 Quiz

This program has three syntax errors.
What are the errors?

```
daat work.NewSalesEmps;
  length First_Name $ 12
    Last_Name $ 18 Job_Title $ 25;
infile 'newemps.csv' dlm=',';
input First_Name $ Last_Name $
        Job_Title $ Salary;
   run;

proc print data=work.NewSalesEmps
   run;

proc means data=work.NewSalesEmps average max;
        class Job_Title;
          var Salary;
          run;
```

18 p103d03

Chapter Review

1. With what do SAS statements usually begin?
   XXX
2. With what do SAS statements always end?
3. What are two methods of commenting?
4. Name four types of syntax errors.
5. How do you save a program?

20

Reading Data

- Reading in of data
  - Input statement
  - Datalines or infile statement
- More advanced direct reading in of data
- Reading in data from SAS datasets (set)
- PROC IMPORT
- Permanent datasets
  - Libname
  - Writing permanent datasets

Reading Data - Datalines

```
Data mydata;
 input name $ gender $ age;
 datalines;
John M 15
Sue F 12
Mary F 14
;
```

Reading Data – Infile and more

```
Data mydata;
 infile 'c:\My Documents\agedata.txt';
 input name $ gender $ age;
```

Useful infile options:

dlm=',' dlm='09'x firstobs=2

Can read in data in various formats
- @@, Fixed length, Missing data, pointers
- See Readdata.sas file

Reading Data Within SAS

- Use the set statement

```
data onlymales;
  set alldata;
  if gender='M';
run;
```

Proc Import

- Allows direct reading of external files
  - Text, Delimited, Excel, Access, etc.
  - There is also an import wizard (helpful the first time around, as it generates SAS code)

```
proc import out=dsname datafile='c:\My Documents\mydata.xlsx'
        dbms=xlsx replace;
        sheet='Raw Data';
        getnames=yes;
run;
```

Permanent Datasets

libname statement

```
libname libref 'c:\My Documents';
```

```
libname mat7500 'c:\Courses\MAT7500';
data mat7500.permname; set tempname;
```

save tempname to a permanent dataset

```
data tempname; set mat7500.permname;
```

reads in permname as tempname (for speed)

Has .sas7bdat extension, keeps variable/dataset attributes

4.05 Poll

During an interactive SAS session, every time that you submit a program you must also resubmit the LIBNAME statement.

- ❍ True
- ❍ False

28

5.02 Poll

If you run the following code…

```
data work.mycustomers;
  set orion.customer;
run;
```

The DATA step reads a temporary SAS data set to create a permanent SAS data set.

- ❍ True
- ❍ False

30

Attributes of Data/Variables

- Variable name requirements
  - Up to 32 characters
  - Starts with letter or _ (underscore)
  - No other special characters allowed
  - Some words (proc, data, work) are reserved
  - Valid names
    dbp12, DiastolicBloodPressure, dbp12
  - Invalid names
    12dbp, dbp 12, dbp*12

Formats and Labels

- Formats and Informats
  - See Informats and Formats.sas
- Dates
  - Number of days since Jan 1, 1960

PROC Contents

- Prints information about the dataset (including size, sorting, etc.)
- Lists all variables
  - Including formats and labels
  - Proc contents varnum; orders by position (as compared to alphabetical by default)
  - Proc contents position; shows both orderings
  - Which one you need depends on context

4.02 Multiple Choice Poll

Which variable type do you think SAS uses to store date values?

a. character
b. numeric

35

4.04 Multiple Answer Poll

Which variable names are valid?

a. `data5mon`
b. `5monthsdata`
c. `data#5`
d. `five months data`
e. `five_months_data`
f. `FiveMonthsData`

37

5.07 Quiz

Which FORMAT statement creates the output?

a.
```
format Birth_Date Hire_Date ddmmyy9.
       Term_Date mmyy7.;
```

b.
```
format Birth_Date Hire_Date ddmmyyyy.
       Term_Date mmmyyyy.;
```

c.
```
format Birth_Date Hire_Date ddmmyy10.
       Term_Date monyy7.;
```

Output

Birth_Date	Hire_Date	Term_Date
21/05/1969	15/10/1992	MAR2007

39

7.06 Quiz (Self-Study)

Copy the following file as phone.csv

```
James Kvarniq,(704) 293-8126,(701) 281-8923
Sandrina Stephano,(919) 871-7830
Cornelia Krahl,(212) 891-3241,(212) 233-5413
Karen Ballinger,(714) 344-4321
Elke Wallstab,(910) 763-5561,(910) 545-3421
```

Run the following code

```
data contacts;
  length Name $ 20 Phone Mobile $ 14;
  infile 'phone.csv' dsd;
  input Name $ Phone $ Mobile $;
run;
proc print data=contacts noobs;  run;
```

Examine the SAS log.
How many input records were read and how many observations were created?

41

7.07 Quiz (Self-Study)

Add the MISSOVER option to the INFILE statement. Submit the program and examine the SAS log. How many input records were read and how many observations were created?

```
data contacts;
   length Name $ 20 Phone Mobile $ 14;
   infile 'phone.csv' dsd;
   input Name $ Phone $ Mobile $;
run;

proc print data=contacts noobs;
run;
```

43

MISSOVER

MISSOVER

prevents an INPUT statement from reading a new input data record if it does not find values in the current input line for all the variables in the statement. Use MISSOVER if the last field or fields might be missing and you want SAS to assign missing values to the corresponding variable.

Name	Phone	Mobile
James Kvarniq	(704) 293-8126	(701) 281-8923
Sandrina Stephano	(919) 871-7830	
Cornelia Krahl	(212) 891-3241	(212) 233-5413
Karen Ballinger	(714) 344-4321	
Elke Wallstab	(910) 763-5561	(910) 545-3421

45

Chapter Review

1. What statement identifies the physical filename of the raw data file to read?
2. What statement describes the arrangement of values in the raw data file?
3. What is the default delimiter when the DLM= option is used?
4. Why would you use a LENGTH statement?

46

8.01 Quiz

What problems will SAS have reading the numeric data **`Salary`** (first highlighted variable) and **`Hire_Date`** (second highlighted variable)?

Partial **`nonsales.csv`**

```
120101,Patrick,Lu,M,163040,Director,AU,18AUG1976,01JUL2003
120104,Kareen,Billington,F,46230,Administration Manager,au,11MAY1954,01JAN1981
120105,Liz,Povey,F,27110,Secretary I,AU,21DEC1974,01MAY1999
120106,John,Hornsey,M,unknown,Office Assistant II,AU,23DEC1944,01JAN1974
120107,Sherie,Sheedy,F,30475,Office Assistant III,AU,01FEB1978,21JAN1953
120108,Gladys,Gromek,F,27660,Warehouse Assistant II,AU,23FEB1984,01AUG2006
120108,Gabriele,Baker,F,26495,Warehouse Assistant I,AU,15DEC1986,01OCT2006
120110,Dennis,Entwisle,M,28615,Warehouse Assistant III,AU,20NOV1949,01NOV1979
120111,Ubaldo,Spillane,M,26895,Security Guard II,AU,23JUL1949,99NOV1978
120112,Ellis,Glattback,F,26550, ,AU,17FEB1969,01JUL1990
120113,Riu,Horsey,F,26870,Security Guard II,AU,10MAY1944,01JAN1974
120114,Jeannette,Buddery,G,31285,Security Manager,AU,08FEB1944,01JAN1974
120115,Hugh,Nicholas,M,2650,Service Assistant I,AU,08MAY1984,01AUG2005
.,Austen,Ralston,M,29250,Service Assistant II,AU,13JUN1959,01FEB1980
120117,Bill,Mccleary,M,31670,Cabinet Maker III,AU,11SEP1964,01APR1986
```

48

8.03 Multiple Choice Poll

Which statements are used to read a delimited raw data file and create a SAS data set?

a. DATA and SET only
b. DATA and INFILE only
c. DATA, SET, and INPUT only
d. DATA, INFILE, and INPUT only

50

Setup for the Poll

- The following program is run:

```
data work.nonsales;
  infile 'nonsales.csv' dlm=',';
  input Employee_ID First $ Last;
run;
```

- To read in the following dataset:

```
120101,Patrick,Lu,M,163040,Director,AU,18AUG1976,01JUL2003
120104,Kareen,Billington,F,46230,Administration Manager,au,11MAY1954,01JAN1981
120105,Liz,Povey,F,27110,Secretary I,AU,21DEC1974,01MAY1999
```

Which statement best describes the invalid data?

a. The data in the raw data file is bad.
b. The programmer incorrectly read the data.

52

8.05 Multiple Choice Poll

Which data requirement cannot be achieved with the PRINT procedure using a WHERE statement?

a. **Employee_ID** must be unique and not missing.
b. **Gender** must have a value of F or M.
c. **Salary** must be in the numeric range of 24000 – 500000.
d. **Job_Title** must not be missing.
e. **Country** must have a value of AU or US.
f. **Birth_Date** value must occur before **Hire_Date** value.
g. **Hire_Date** must have a value of 01/01/1974 or later.

54

8.07 Quiz

Which variable can be used to specifically identify the observations with invalid salary values?

Obs	Employee_ID	Gender	Salary	Job_Title	Country	Birth_Date	Hire_Date
2	120104	F	46230	Administration Manager	au	11/05/1954	01/01/1981
4	120106	M		Office Assistant II	AU	23/12/1944	01/01/1974
5	120107	F	30475	Office Assistant III	AU	01/02/1978	21/01/1953
9	120111	M	26895	Security Guard II	AU	23/07/1949	.
10	120112	F	26550		AU	17/02/1969	01/07/1990
12	120114	G	31285	Security Manager	AU	08/02/1944	01/01/1974
13	120115	M	2650	Service Assistant I	AU	08/05/1984	01/08/2005
14	.	M	29250	Service Assistant II	AU	13/06/1959	01/02/1980
20	120191	F	2401	Trainee	AU	17/01/1959	01/01/2003
84	120695	M	28180	Warehouse Assistant II	au	13/07/1964	01/07/1989
87	120698	M	26160	Warehouse Assistant I	au	17/05/1954	01/08/1976
101	120723		33950	Corp. Comm. Specialist II	US	10/08/1949	01/01/1974
125	120747	F	43590	Financial Controller I	us	20/06/1974	01/08/1995
197	120994	F	31645	Office Administrator I	us	16/06/1974	01/11/1994
200	120997	F	27420	Shipping Administrator I	us	21/11/1974	01/09/1996
214	121011	M	25735	Service Assistant I	US	11/03/1944	01/01/1968

56

Combining Data

- PROC Sort
- Set (part of DATA Step)
  - "Appending" and "Concatenating"
- Merge (part of DATA Step)
- Nonmatching data in set and merge
- Issues with merging data

Proc Sort

- Sort the data by one or more variables
  proc sort data=mydata; by ID;
- Can sort in descending order
  by descending ID;
- Can remove duplicates (I recommend creating a new dataset with out=)
  proc sort data=mydata out=newdata nodup; by ID;
  removes all duplicates
  proc sort data=mydata out=newdata nodupkey; by ID;
  removes subsequent duplicate records of by var(s)
- Necessary for some PROCs

Nodup vs. Nodupkey

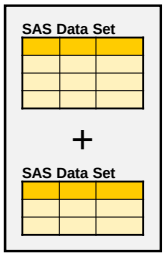
- Consider the following data:

ID	Var1
001	100
001	100
001	200
002	100
002	200

- Using nodup with by ID removes the second obs
  - Useful for checking data
- Using nodupkey with by ID removes the second, third, and fifth obs
  - Useful for multilevel/nested data

Appending and Concatenating (SET Statement)

Appending and concatenating involves combining SAS data sets, one after the other, into a single SAS data set.

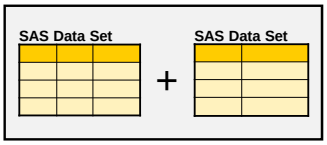


- *Appending* adds the observations in the second data set directly to the end of the original data set.
- *Concatenating* copies all observations from the first data set and then copies all observations from one or more successive data sets into a new data set.

61

Merging

Merging involves combining observations from two or more SAS data sets into a single observation in a new SAS data set.



Observations can be merged based on their positions in the original data sets or merged by one or more common variables.

62

Example: Appending a Data Set

One data set is appended to a master data set.

Emps

First	Gender	HireYear
Stacey	F	2006
Gloria	F	2007
James	M	2007

Emps2008

First	Gender	HireYear
Brett	M	2008
Renee	F	2008

Emps

First	Gender	HireYear
Stacey	F	2006
Gloria	F	2007
James	M	2007
Brett	M	2008
Renee	F	2008

63

Example: Concatenating Data Sets

Two data sets are concatenated to create a new data set.

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

EmpsAll1

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark
Pierre	M	France
Sophie	F	France

64

Compiling/Executing a SET statement in SAS

Concatenate **EmpsDK** and **EmpsFR** to create a new data set named **EmpsAll1**.

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

The data sets contain the same variables.

```
data EmpsAll1;
   set EmpsDK EmpsFR;
run;
```

65 p110d02

Compilation

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

```
data EmpsAll1;
   set EmpsDK EmpsFR;
run;
```

PDV

First	Gender	Country

EmpsAll1

First	Gender	Country

66 ...

Execution

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

```
data EmpsAll1;
   set EmpsDK EmpsFR;
run;
```

Initialize PDV

PDV

First	Gender	Country

EmpsAll1

First	Gender	Country

67 ...

Execution

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

```
data EmpsAll1;
   set EmpsDK EmpsFR;
run;
```

PDV

First	Gender	Country
Lars	M	Denmark

EmpsAll1

First	Gender	Country

Execution

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

```
data EmpsAll1;
   set EmpsDK EmpsFR;
run;
```

Implicit OUTPUT;
Implicit RETURN;

PDV

First	Gender	Country
Lars	M	Denmark

EmpsAll1

First	Gender	Country
Lars	M	Denmark

Execution

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

```
data EmpsAll1;
   set EmpsDK EmpsFR;
run;
```

PDV

First	Gender	Country
Kari	F	Denmark

EmpsAll1

First	Gender	Country
Lars	M	Denmark

Execution

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

```
data EmpsAll1;
   set EmpsDK EmpsFR;
run;
```

Implicit OUTPUT;
Implicit RETURN;

PDV

First	Gender	Country
Kari	F	Denmark

EmpsAll1

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark

Execution

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

```
data EmpsAll1;
   set EmpsDK EmpsFR;
run;
```

PDV

First	Gender	Country
Jonas	M	Denmark

EmpsAll1

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark

Execution

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

```
data EmpsAll1;
   set EmpsDK EmpsFR;
run;
```

Implicit OUTPUT;
Implicit RETURN;

PDV

First	Gender	Country
Jonas	M	Denmark

EmpsAll1

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

Execution

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EOF

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

```
data EmpsAll1;
   set EmpsDK EmpsFR;
run;
```

PDV

First	Gender	Country
Jonas	M	Denmark

EmpsAll1

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

74 ...

Execution

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

```
data EmpsAll1;
   set EmpsDK EmpsFR;
run;
```

Reinitialize PDV

PDV

First	Gender	Country

EmpsAll1

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

75 ...

Execution

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

```
data EmpsAll1;
   set EmpsDK EmpsFR;
run;
```

PDV

First	Gender	Country
Pierre	M	France

EmpsAll1

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

76 ...

Execution

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

```
data EmpsAll1;
   set EmpsDK EmpsFR;
run;
```

Implicit OUTPUT;
Implicit RETURN;

PDV

First	Gender	Country
Pierre	M	France

EmpsAll1

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark
Pierre	M	France

77 ...

Execution

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

```
data EmpsAll1;
   set EmpsDK EmpsFR;
run;
```

PDV

First	Gender	Country
Sophie	F	France

EmpsAll1

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark
Pierre	M	France

78 ...

Execution

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

```
data EmpsAll1;
   set EmpsDK EmpsFR;
run;
```

Implicit OUTPUT;
Implicit RETURN;

PDV

First	Gender	Country
Sophie	F	France

EmpsAll1

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark
Pierre	M	France
Sophie	F	France

79 ...

Execution

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

EOF

```
data EmpsAll1;
   set EmpsDK EmpsFR;
run;
```

PDV

First	Gender	Country
Sophie	F	France

EmpsAll1

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark
Pierre	M	France
Sophie	F	France

80

Example: Merging Data Sets

Two data sets are merged to create a new data set.

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneH

EmpID	Phone
121150	+61(2)5555-1793
121151	+61(2)5555-1849
121152	+61(2)5555-1665

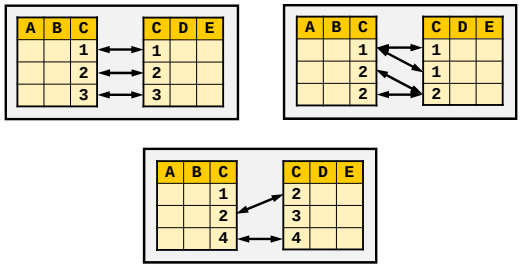
EmpsAUH

First	Gender	EmpID	Phone
Togar	M	121150	+61(2)5555-1793
Kylie	F	121151	+61(2)5555-1849
Birin	M	121152	+61(2)5555-1665

81

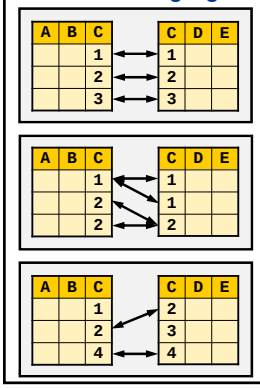
Match-Merging

Match-merging combines observations from two or more SAS data sets into a single observation in a new data set based on the values of one or more common variables.



82

Match-Merging



One-to-One

A single observation in one data set is related to one and only one observation from another data set based on the values of one or more selected variables.

One-to-Many or Many-to-One

A single observation in one data set is related to more than one observation from another data set based on the values of one or more selected variables and vice versa.

Nonmatches

At least one single observation in one data set is unrelated to any observation from another data set based on the values of one or more selected variables.

One-to-One Merge

Merge **EmpsAU** and **PhoneH** by **EmpID** to create a new data set named **EmpsAUH**.

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneH

EmpID	Phone
121150	+61(2)5555-1793
121151	+61(2)5555-1849
121152	+61(2)5555-1665

The data sets must be sorted by **EmpID**.

```
data EmpsAUH;
   merge EmpsAU PhoneH;
   by EmpID;
run;
```

84 p110d05

One-to-Many Merge

Merge **EmpsAU** and **PhoneHW** by **EmpID** to create a new data set named **EmpsAUHW**.

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneHW

EmpID	Type	Phone
121150	Home	+61(2)5555-1793
121150	Work	+61(2)5555-1794
121151	Home	+61(2)5555-1849
121151	Work	+61(2)5555-1850
121152	Home	+61(2)5555-1665
121152	Work	+61(2)5555-1666

```
data EmpsAUHW;
   merge EmpsAU PhoneHW;
   by EmpID;
run;
```

The data sets are sorted by **EmpID**.

85 p110d06

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneHW

EmpID	Type	Phone
121150	Home	+61(2)5555-1793
121150	Work	+61(2)5555-1794
121151	Home	+61(2)5555-1849
121151	Work	+61(2)5555-1850
121152	Home	+61(2)5555-1665
121152	Work	+61(2)5555-1666

```
data EmpsAUHW;
   merge EmpsAU
   by EmpID;
run;
```

Initialize PDV

PDV

First	Gender	EmpID	Type	Phone
		.		

86 ...

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneHW

EmpID	Type	Phone
121150	Home	+61(2)5555-1793
121150	Work	+61(2)5555-1794
121151	Home	+61(2)5555-1849
121151	Work	+61(2)5555-1850
121152	Home	+61(2)5555-1665
121152	Work	+61(2)5555-1666

```
data EmpsAUHW;
   merge EmpsAU PhoneHW;
   by EmpID;
run;
```

Do the **EmpID**s match?

Yes

PDV

First	Gender	EmpID	Type	Phone
		.		

87 ...

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneHW

EmpID	Type	Phone
121150	Home	+61(2)5555-1793
121150	Work	+61(2)5555-1794
121151	Home	+61(2)5555-1849
121151	Work	+61(2)5555-1850
121152	Home	+61(2)5555-1665
121152	Work	+61(2)5555-1666

```
data EmpsAUHW;
   merge EmpsAU PhoneHW;
   by EmpID;
run;
```

Reads one observation from each matching data set

PDV

First	Gender	EmpID	Type	Phone
Togar	M	121150	Home	+61(2)5555-1793

88 ...

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneHW

EmpID	Type	Phone
121150	Home	+61(2)5555-1793
121150	Work	+61(2)5555-1794
121151	Home	+61(2)5555-1849
121151	Work	+61(2)5555-1850
121152	Home	+61(2)5555-1665
121152	Work	+61(2)5555-1666

```
data EmpsAUHW;
   merge EmpsAU PhoneHW;
   by EmpID;
run;
```

Implicit OUTPUT;
Implicit RETURN;

PDV

First	Gender	EmpID	Type	Phone
Togar	M	121150	Home	+61(2)5555-1793

89 ...

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneHW

EmpID	Type	Phone
121150	Home	+61(2)5555-1793
121150	Work	+61(2)5555-1794
121151	Home	+61(2)5555-1849
121151	Work	+61(2)5555-1850
121152	Home	+61(2)5555-1665
121152	Work	+61(2)5555-1666

```
data EmpsAUHW;
   merge EmpsAU PhoneHW;
   by EmpID;
run;
```

Do the **EmpID**s match?

No

PDV

First	Gender	EmpID	Type	Phone
Togar	M	121150	Home	+61(2)5555-1793

90 ...

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneHW

EmpID	Type	Phone
121150	Home	+61(2)5555-1793
121150	Work	+61(2)5555-1794
121151	Home	+61(2)5555-1849
121151	Work	+61(2)5555-1850
121152	Home	+61(2)5555-1665
121152	Work	+61(2)5555-1666

```
data EmpsAUHW;
   merge EmpsAU PhoneHW;
   by EmpID;
run;
```

Is either **EmpID** the same as the **EmpID** currently in the PDV?

Yes

PDV

First	Gender	EmpID	Type	Phone
Togar	M	121150	Home	+61(2)5555-1793

91 ...

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneHW

EmpID	Type	Phone
121150	Home	+61(2)5555-1793
121150	Work	+61(2)5555-1794
121151	Home	+61(2)5555-1849
121151	Work	+61(2)5555-1850
121152	Home	+61(2)5555-1665
121152	Work	+61(2)5555-1666

```
data EmpsAUHW;
   merge EmpsAU PhoneHW;
   by EmpID;
run;
```

Reads the observation from the appropriate data set

PDV

First	Gender	EmpID	Type	Phone
Togar	M	121150	Work	+61(2)5555-1794

92 ...

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneHW

EmpID	Type	Phone
121150	Home	+61(2)5555-1793
121150	Work	+61(2)5555-1794
121151	Home	+61(2)5555-1849
121151	Work	+61(2)5555-1850
121152	Home	+61(2)5555-1665
121152	Work	+61(2)5555-1666

```
data EmpsAUHW;
   merge EmpsAU PhoneHW;
   by EmpID;
run;
```

Implicit OUTPUT;
Implicit RETURN;

PDV

First	Gender	EmpID	Type	Phone
Togar	M	121150	Work	+61(2)5555-1794

93 ...

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneHW

EmpID	Type	Phone
121150	Home	+61(2)5555-1793
121150	Work	+61(2)5555-1794
121151	Home	+61(2)5555-1849
121151	Work	+61(2)5555-1850
121152	Home	+61(2)5555-1665
121152	Work	+61(2)5555-1666

```
data EmpsAUHW;
   merge EmpsAU PhoneHW;
   by EmpID;
run;
```

Do the **EmpID**s match?

Yes

PDV

First	Gender	EmpID	Type	Phone
Togar	M	121150	Work	+61(2)5555-1794

94 ...

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneHW

EmpID	Type	Phone
121150	Home	+61(2)5555-1793
121150	Work	+61(2)5555-1794
121151	Home	+61(2)5555-1849
121151	Work	+61(2)5555-1850
121152	Home	+61(2)5555-1665
121152	Work	+61(2)5555-1666

```
data EmpsAUHW;
   merge EmpsAU PhoneHW;
   by EmpID;
run;
```

Is the **EmpID** the same as the **EmpID** currently in the PDV?

No

PDV

First	Gender	EmpID	Type	Phone
Togar	M	121150	Work	+61(2)5555-1794

95 ...

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneHW

EmpID	Type	Phone
121150	Home	+61(2)5555-1793
121150	Work	+61(2)5555-1794
121151	Home	+61(2)5555-1849
121151	Work	+61(2)5555-1850
121152	Home	+61(2)5555-1665
121152	Work	+61(2)5555-1666

```
data EmpsAUHW;
   merge EmpsAU PhoneHW;
   by EmpID;
run;
```

Reinitialize PDV

PDV

First	Gender	EmpID	Type	Phone
		.		

96 ...

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneHW

EmpID	Type	Phone
121150	Home	+61(2)5555-1793
121150	Work	+61(2)5555-1794
121151	Home	+61(2)5555-1849
121151	Work	+61(2)5555-1850
121152	Home	+61(2)5555-1665
121152	Work	+61(2)5555-1666

```
data EmpsAUHW;
   merge EmpsAU PhoneHW;
   by EmpID;
run;
```

Reads one observation from each matching data set

PDV

First	Gender	EmpID	Type	Phone
Kylie	F	121151	Home	+61(2)5555-1849

97 ...

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneHW

EmpID	Type	Phone
121150	Home	+61(2)5555-1793
121150	Work	+61(2)5555-1794
121151	Home	+61(2)5555-1849
121151	Work	+61(2)5555-1850
121152	Home	+61(2)5555-1665
121152	Work	+61(2)5555-1666

```
data EmpsAUHW;
   merge EmpsAU PhoneHW;
   by EmpID;
run;
```

Implicit OUTPUT;
Implicit RETURN;

PDV

First	Gender	EmpID	Type	Phone
Kylie	F	121151	Home	+61(2)5555-1849

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneHW

EmpID	Type	Phone
121150	Home	+61(2)5555-1793
121150	Work	+61(2)5555-1794
121151	Home	+61(2)5555-1849
121151	Work	+61(2)5555-1850
121152	Home	+61(2)5555-1665
121152	Work	+61(2)5555-1666

```
data EmpsAUHW;
   merge EmpsAU PhoneHW;
   by EmpID;
run;
```

Do the **EmpID**s match?

No

PDV

First	Gender	EmpID	Type	Phone
Kylie	F	121151	Home	+61(2)5555-1849

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneHW

EmpID	Type	Phone
121150	Home	+61(2)5555-1793
121150	Work	+61(2)5555-1794
121151	Home	+61(2)5555-1849
121151	Work	+61(2)5555-1850
121152	Home	+61(2)5555-1665
121152	Work	+61(2)5555-1666

```
data EmpsAUHW;
   merge EmpsAU PhoneHW;
   by EmpID;
run;
```

Is either **EmpID** the same as the **EmpID** currently in the PDV?

Yes

PDV

First	Gender	EmpID	Type	Phone
Kylie	F	121151	Home	+61(2)5555-1849

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneHW

EmpID	Type	Phone
121150	Home	+61(2)5555-1793
121150	Work	+61(2)5555-1794
121151	Home	+61(2)5555-1849
121151	Work	+61(2)5555-1850
121152	Home	+61(2)5555-1665
121152	Work	+61(2)5555-1666

```
data EmpsAUHW;
   merge EmpsAU PhoneHW;
   by EmpID;
run;
```

Reads the observation from the appropriate data set

PDV

First	Gender	EmpID	Type	Phone
Kylie	F	121151	Work	+61(2)5555-1850

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneHW

EmpID	Type	Phone
121150	Home	+61(2)5555-1793
121150	Work	+61(2)5555-1794
121151	Home	+61(2)5555-1849
121151	Work	+61(2)5555-1850
121152	Home	+61(2)5555-1665
121152	Work	+61(2)5555-1666

```
data EmpsAUHW;
   merge EmpsAU PhoneHW;
   by EmpID;
run;
```

Implicit OUTPUT;
Implicit RETURN;

PDV

First	Gender	EmpID	Type	Phone
Kylie	F	121151	Work	+61(2)5555-1850

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneHW

EmpID	Type	Phone
121150	Home	+61(2)5555-1793
121150	Work	+61(2)5555-1794
121151	Home	+61(2)5555-1849
121151	Work	+61(2)5555-1850
121152	Home	+61(2)5555-1665
121152	Work	+61(2)5555-1666

```
data EmpsAUHW;
   merge EmpsAU Ph
   by EmpID;
run;
```

Continue until EOF on both data sets

PDV

First	Gender	EmpID	Type	Phone
Kylie	F	121151	Work	+61(2)5555-1850

Final Results

EmpsAUHW

First	Gender	EmpID	Type	Phone
Togar	M	121150	Home	+61(2)5555-1793
Togar	M	121150	Work	+61(2)5555-1794
Kylie	F	121151	Home	+61(2)5555-1849
Kylie	F	121151	Work	+61(2)5555-1850
Birin	M	121152	Home	+61(2)5555-1665
Birin	M	121152	Work	+61(2)5555-1666

Nonmatches Merge

Merge **EmpsAU** and **PhoneC** by **EmpID** to create a new data set named **EmpsAUC**.

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneC

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

The data sets are sorted by **EmpID**.

```
data EmpsAUC;
   merge EmpsAU PhoneC;
   by EmpID;
run;
```


Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneC

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data EmpsAUC;
   merge EmpsA
   by EmpID;
run;
```

Initialize PDV

PDV

First	Gender	EmpID	Phone
		.	

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneC

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data EmpsAUC;
   merge EmpsAU PhoneC;
   by EmpID;
run;
```

Do the **EmpID**s match?

Yes

PDV

First	Gender	EmpID	Phone
		.	

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneC

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data EmpsAUC;
   merge EmpsAU PhoneC;
   by EmpID;
run;
```

Reads one observation from each matching data set

PDV

First	Gender	EmpID	Phone
Togar	M	121150	+61(2)5555-1795

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneC

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data EmpsAUC;
   merge EmpsAU PhoneC;
   by EmpID;
run;
```

Implicit OUTPUT;
Implicit RETURN;

PDV

First	Gender	EmpID	Phone
Togar	M	121150	+61(2)5555-1795

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneC

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data EmpsAUC;
   merge EmpsAU PhoneC;
   by EmpID;
run;
```

Do the **EmpID**s match?

No

PDV

First	Gender	EmpID	Phone
Togar	M	121150	+61(2)5555-1795

110 ...

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneC

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data EmpsAUC;
   merge EmpsAU PhoneC;
   by EmpID;
run;
```

Is either **EmpID** the same as the **EmpID** currently in the PDV?

No

PDV

First	Gender	EmpID	Phone
Togar	M	121150	+61(2)5555-1795

111 ...

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneC

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data EmpsAUC;
   merge EmpsAU PhoneC;
   by EmpID;
run;
```

Reinitialize PDV

PDV

First	Gender	EmpID	Phone
		.	

112 ...

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneC

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data EmpsAUC;
   merge EmpsAU PhoneC;
   by EmpID;
run;
```

Which **EmpID** sequentially comes first?

121151

PDV

First	Gender	EmpID	Phone
		.	

113 ...

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneC

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data EmpsAUC;
   merge EmpsAU PhoneC;
   by EmpID;
run;
```

Reads the observation from the **EmpID** that sequentially comes first

PDV

First	Gender	EmpID	Phone
Kylie	F	121151	

114 ...

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneC

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data EmpsAUC;
   merge EmpsAU PhoneC;
   by EmpID;
run;
```

Implicit OUTPUT;
Implicit RETURN;

PDV

First	Gender	EmpID	Phone
Kylie	F	121151	

115 ...

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneC

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data EmpsAUC;
   merge EmpsAU PhoneC;
   by EmpID;
run;
```

Do the **EmpID**s match?

Yes

PDV

First	Gender	EmpID	Phone
Kylie	F	121151	

116 ...

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneC

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data EmpsAUC;
   merge EmpsAU PhoneC;
   by EmpID;
run;
```

Is either **EmpID** the same as the **EmpID** currently in the PDV?

No

PDV

First	Gender	EmpID	Phone
Kylie	F	121151	

117 ...

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneC

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data EmpsAUC;
   merge EmpsAU PhoneC;
   by EmpID;
run;
```

Reinitialize PDV

PDV

First	Gender	EmpID	Phone
		.	

118 ...

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneC

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data EmpsAUC;
   merge EmpsAU PhoneC;
   by EmpID;
run;
```

Reads one observation from each matching data set

PDV

First	Gender	EmpID	Phone
Birin	M	121152	+61(2)5555-1667

119 ...

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

PhoneC

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data EmpsAUC;
   merge EmpsAU PhoneC;
   by EmpID;
run;
```

Implicit OUTPUT;
Implicit RETURN;

PDV

First	Gender	EmpID	Phone
Birin	M	121152	+61(2)5555-1667

120 ...

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

EOF

PhoneC

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data EmpsAUC;
   merge EmpsAU PhoneC;
   by EmpID;
run;
```

Is the **EmpID** the same as the **EmpID** currently in the PDV?

No

PDV

First	Gender	EmpID	Phone
Birin	M	121152	+61(2)5555-1667

121 ...

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

EOF

PhoneC

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data EmpsAUC;
   merge EmpsAU PhoneC;
   by EmpID;
run;
```

Reinitialize PDV

PDV

First	Gender	EmpID	Phone
		.	

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

EOF

PhoneC

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data EmpsAUC;
   merge EmpsAU PhoneC;
   by EmpID;
run;
```

Reads the observation from the appropriate data set

PDV

First	Gender	EmpID	Phone
		121153	+61(2)5555-1348

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

EOF

PhoneC

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data EmpsAUC;
   merge EmpsAU PhoneC;
   by EmpID;
run;
```

Implicit OUTPUT;
Implicit RETURN;

PDV

First	Gender	EmpID	Phone
		121153	+61(2)5555-1348

Execution

EmpsAU

First	Gender	EmpID
Togar	M	121150
Kylie	F	121151
Birin	M	121152

EOF

PhoneC

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

EOF

```
data EmpsAUC;
   merge EmpsAU PhoneC;
   by EmpID;
run;
```

PDV

First	Gender	EmpID	Phone
		121153	+61(2)5555-1348

Final Results

EmpsAUC

First	Gender	EmpID	Phone
Togar	M	121150	+61(2)5555-1795
Kylie	F	121151	
Birin	M	121152	+61(2)5555-1667
		121153	+61(2)5555-1348

The final results include matches and nonmatches.

- Matches are observations that contain data from both input data sets.
- Nonmatches are observations that contain data from only one input data set.

Differing Data in Set/Merge

- Variables missing in some datasets being set will be missing
- The rename= statement can be used to combine datafiles with different names

  Ex: data1 has “sex” but data2 has “gender”

  ```
  data combined;
      set data1 (rename=(sex=gender)) data2;
  ```

- Observations missing for some datasets being merged will be set to missing
- Multiple observations by the merge key will be matched multiple times (which may or may not be the desired outcome)

Issues with Merging Data

- Be careful with merges!
  - Overwriting datasets
  - Incorrect merges (without by statement)
  - We will learn a trick with merges later (in=a)
- Formats
  - SAS will complain about combining data with different formats (more on this later)

10.01 Quiz

Which method (appending, concatenating, or merging) should be used for the given business scenario?

	Business Scenario	Method
1	The **JanSales**, **FebSales**, and **MarSales** data sets need to be combined to create the **Qtr1Sales** data set.	
2	The **Sales** data set needs to be combined with the **Target** data set by **month** to compare the sales data to the target data.	
3	The **OctSales** data set needs to be added to the **YTD** data set.	

129

10.05 Quiz

How many variables will be in **EmpsAll2** after concatenating **EmpsCN** and **EmpsJP**?

EmpsCN

First	Gender	Country
Chang	M	China
Li	M	China
Ming	F	China

EmpsJP

First	Gender	Region
Cho	F	Japan
Tomi	M	Japan

```
data EmpsAll2;
   set EmpsCN EmpsJP;
run;
```

131

10.06 Quiz

Which statement has correct syntax?

a.
```
set EmpsCN(rename(Country=Location))
    EmpsJP(rename(Region=Location));
```

b.
```
set EmpsCN(rename=(Country=Location))
    EmpsJP(rename=(Region=Location));
```

c.
```
set EmpsCN rename=(Country=Location)
    EmpsJP rename=(Region=Location);
```

133

10.08 Quiz

Which step is sorting the observations in a SAS data set and overwriting the same SAS data set?

a.
```
proc sort data=work.EmpsAU
          out=work.sorted;
   by First;
run;
```

b.
```
proc sort data=work.EmpsAU
          out=orion.EmpsAU;
   by First;
run;
```

c.
```
proc sort data=work.EmpsAU;
   by First;
run;
```

135

10.09 Multiple Choice Poll

Look again at the sales.xls dataset, including both the US and Australia sheets. Considering the following statement:

Proc sort;
 by gender descending employee_ID;

What is the **Employee_ID** value for the first observation in the sorted data set?

a. Female 120102
b. Male 120121
c. Female 121144
d. Male 121145

137