

## Lecture 4

1. Comments on HW2
2. Data set options
3. First and Last variables
4. Finding unmatched or discrepant observations in a merge
5. Subsetting IF
6. Input data checking
7. Proc Insight: plots and histograms
8. Proc Univariate: checking data within SAS

1

### Data Set Options (*LSB* §6.11)

Data set options change the way SAS reads a data set.

Option(s) go in parentheses after the data set name:

```
Data new_data;  
    set old_data (option) ;
```

1. Options that work on rows (observations):

**FIRSTOBS** = tells SAS where to start processing the data

**OBS** = tells SAS where to stop processing the data

Need **FIRSTOBS** < **OBS** to read any observations

2

Very helpful in printing part of a long dataset:

```
Proc Print data = ph6470.child_IQ (obs = 5) ;
```

Obs	ID	child_IQ	mom_HS_ grad	mom_age	mom_IQ	male
1	1	65	1	27	121	1
2	2	98	1	25	89	1
3	3	85	1	27	115	0
4	4	83	1	25	99	1
5	5	115	1	27	93	0

3

2. Options that work on variables (columns):

**DROP** = *variable list*: specifies *variables* to be excluded

Data B;

```
set A (drop = First_Name Last_Name); no commas
```

**KEEP** = *variable list*: specifies *variables* to be included

```
RENAME = ( oldname1=newname1 oldname2=newname2 )
```

changes the names of variables oldname1 and oldname2.

Data B;

```
set A (rename = (mask_patient = ID)); double parentheses
```

4

**IN** = *new-variable name*:

Creates an indicator variable that tells whether the current observation has variables from this data set or not.

This variable only exists during the dataset: if you print the data it won't appear. To keep it, create a new variable equal to the IN-variable.

Merge A and B, but keep track of which datasets contribute to each row.

Data A			Data B		
id	color	mass	id	mass	pH
12	orange	3650	13	11267	7.8
13	blue	3877	14	3568	8.2
15	yellow	4103	15	4103	5.1

5

```
data D;
  merge A (in = in_a)  B (in = in_b);
  by id;
  save_in_a = in_a;
  save_in_b = in_b;
  in_both = (in_a=1 AND in_b=1);
```

```
Proc Print data=D;
```

Obs	id	color	mass	pH	save_ in_a	save_ in_b	in_both
1	12	orange	3650	.	1	0	0
2	13	blue	11267	7.8	1	1	1
3	14		3568	8.2	0	1	0
4	15	yellow	4103	5.1	1	1	1

6

## More on SET and MERGE (LSB Chapter 6)

merge BY *variable* and set BY *variable*, create two secret variables (LSB §6.12) :

`first.BYvariable` = 1 for the first observation with this value of variable, = 0 otherwise

`last.BYvariable` = 1 for the last observation with this value of variable, = 0 otherwise

Useful when the data has multiple observations for each ID.

Data F		
id	visit	weight
101	1	145
101	2	149
101	3	152
102	1	181
102	2	176

7

```
data H;  
  set F;  
  by id;  
  first_id = first.id; make a copy to save these variables  
  last_id = last.id;
```

```
Proc Print data=H;
```

Obs	id	visit	weight	first_id	last_id
1	101	1	145	1	0
2	101	2	149	0	0
3	101	3	152	0	1
4	102	1	181	1	0
5	102	2	176	0	1

8

To get average weight over visits by patient:

start the sum with `first.BYvariable = 1`

and calculate the average at `last.BYvariable = 1`.

Another secret variable is: `_N_` = the number of times SAS has gone through the whole data step

Like `_N_`, these variables only exist during the data step that creates them. To keep them, make a new variable equal to the one you want to keep.

9

### Merging with missing and duplicate values

Merge two spreadsheets with clinical data recorded at each visit.

Data E			Data F		
id	visit	DBP	id	visit	weight
101	1	77	101	1	145
101	2	75	101	2	149
.	3	80	101	3	152
102	1	71	102	1	181
102	2	74	102	2	176
102	3	68			

One extra observation in dataset E.

```
proc sort data=E; by id; sort by ID first
```

```
proc sort data=F; by id;
```

```
data G;
```

```
merge E F;
```

```
by id;
```

```
proc print data=G;
```

11

Obs	id	visit	dbp	weight
1	.	3	80	.
2	101	1	77	145
3	101	2	75	149
4	101	3	75	152
5	102	1	71	181
6	102	2	74	176
7	102	3	68	176

*where did these come from?*

Data E			Data F		
id	visit	DBP	id	visit	weight
101	1	77	101	1	145
101	2	75	101	2	149
.	3	80	101	3	152
102	1	71	102	1	181
102	2	74	102	2	176
102	3	68			

12

Warning in the log file:

```
394 data g;  
395 merge e f;  
396 by id;
```

**NOTE:**

MERGE statement has more than one data set with repeats of BY values.

Within each ID, SAS is merging *without any matching variable*.

Don't merge by only one variable if it has missing values or duplicates.

13

Sort by as many variables as needed to identify each observation.

Then merge by all of them.

Here, ID and visit are enough.

Data E			Data F		
id	visit	DBP	id	visit	weight
101	1	77	101	1	145
101	2	75	101	2	149
.	3	80	101	3	152
102	1	71	102	1	181
102	2	74	102	2	176
102	3	68			

14

```
proc sort data=E; by id visit;
proc sort data=F; by id visit;
```

```
data K;
  merge E F;
  by id visit;
```

```
proc print data=K;
```

```
Obs      id      visit      dbp      weight
  1         .         3         80         .
  2       101         1         77        145
  3       101         2         75        149
  4       101         3          .        152
  5       102         1         71        181
  6       102         2         74        176
  7       102         3         68          .
```

15

### Finding unmatched observations in a MERGE

An “unmatched observation” appears in only one of the data sets being merged:

Data A			Data B		
id	color	mass	id	mass	pH
12	orange	3650	13	11267	7.8
13	blue	3877	14	3568	8.2
15	yellow	4103	15	4103	5.1

We would like to merge A with B by `id`, and then get a list of the unmatched observations that tells which data set they came from.

We use a dataset option (`IN = variable`) to create an indicator that labels the source.

```
Proc Sort data = A; by id; first SORT by ID
Proc Sort data = B; by id;
```

```
data G;
  merge A(in = in_a) B(in = in_b) ;
  by id;
  from_a = in_a ; save these variables by renaming them
  from_b = in_b ;
Proc Print;
```

The whole merged data set:

Obs	id	color	mass	pH	from_a	from_b
1	12	orange	3650	.	1	0
2	13	blue	11267	7.8	1	1
3	14		3568	8.2	0	1
4	15	yellow	4103	5.1	1	1

17

To get a list of the unmatched observations:

```
data G;
  merge A (in = in_a) B (in = in_b) ;
  by id;
  if (in_a NE in_b); subsetting IF statement
  from_a = in_a ; save these to identify source
  from_b = in_b ;
```

```
Proc Print data = G;
```

Obs	id	color	mass	pH	from_a	from_b
1	12	orange	3650	.	1	0
2	14		3568	8.2	0	1

How could we get this list with all the A observations first?

18

## Subsetting IF (LSB §3.7)

To select certain observations (rows) from a data set :

```
Data M; output data
  set L; input data
  IF (statement);
```

Each row of L, SAS tests whether (*statement*) is true for that row.

If true, observation is kept in M.

If false, observation is deleted from M.

Subsetting IF has no effect on input data L.

19

### Finding discrepant values for the same observation in a MERGE

Data A			Data B		
id	color	mass	id	mass	pH
12	orange	3650	13	11267	7.8
13	blue	3877	14	3568	8.2
15	yellow	4103	15	4103	5.1

We would like to merge A with B, and then get a list of the observations that have discrepant values for `mass`.

20

What about this?

```
Proc Sort data = A;  by id;  
Proc Sort data = B;  by id;
```

```
data I;  
  merge A  B ;  
  by id;  
  if (mass NE mass);
```

21

The whole merged data set before the subsetting IF:

Obs	id	color	mass	pH
1	12	orange	3650	.
2	13	blue	11267	7.8
3	14		3568	8.2
4	15	yellow	4103	5.1

When is (mass NE mass) true?

22

We need to keep separate mass variables from each data set:

```
Proc Sort data = A; by id;
```

```
Proc Sort data = B; by id;
```

```
data C;
```

```
merge A(rename = (mass = mass_a)) B(rename = (mass = mass_b)) ;
```

```
by id;
```

```
Proc Print;
```

Obs	id	color	mass_a	mass_b	pH
1	12	orange	3650	.	.
2	13	blue	3877	11267	7.8
3	14		.	3568	8.2
4	15	yellow	4103	4103	5.1

23

List of observations with discrepant values:

```
data C;
```

```
merge A(rename = (mass = mass_a)) B(rename = (mass = mass_b))
```

```
by id;
```

```
if (mass_a NE mass_b) ;
```

```
Proc Print;
```

Obs	id	color	mass_a	mass_b	pH
1	12	orange	3650	.	.
2	13	blue	3877	11267	7.8
3	14		.	3568	8.2

24

## Phases of Data Checking

1. SAS recognizes several types for data (character, numeric, date) and each of these has various different formats.

SAS distinguishes three types of data, and has different methods for dealing with each type. Each variable (column) is assigned to one of these types:

- Numeric data: integer or floating point
- Character data: any combination of letters, numbers, spaces, punctuation
- Date-time data: calendar dates and times of day

25

The first phase of checking asks:

*Did SAS read the data as the correct type?*

*Did SAS read the correct number of observations and variables?*

Proc Contents answers both these questions.

```
proc contents data = PH6470.child_iq;
```

26

The CONTENTS Procedure

Data Set Name	PH6470.CHILD_IQ	Observations	434
Member Type	DATA	Variables	6
Engine	V9	Indexes	0
Created	Thu, Sep 10, 2009 11:55:16 AM	Observation Length	48

Filename	C:\Documents and Settings\Administrator\Desktop\SAS Class\child_iq.sas7bdat		
Release Created	9.0201M0		
Host Created	XP_PRO		

[other stuff]

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Label
1	ID	Num	8	ID
2	child_IQ	Num	8	child IQ
6	male	Num	8	male
3	mom_HS_grad	Num	8	mom HS grad
5	mom_IQ	Num	8	mom IQ
4	mom_age	Num	8	mom age

27

Here is data from HW2 which contains character and date-time variables:

The CONTENTS Procedure

Data Set Name	PUBH.HW2_DEMOGRAPHICS	Observations	899
Member Type	DATA	Variables	5
Engine	V9	Indexes	0
Created	Monday, September 22, 2008 06:18:02 AM	Observation Length	40
Last Modified	Monday, September 22, 2008 06:18:02 AM	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	YES

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Informat	Label
3	birthdate	Num	8	DATE9.	DATE9.	birthdate
2	female	Num	8			female
5	rand_date	Num	8	DATE9.		
1	subject	Num	8			subject
4	treatment	Char	1			

28

2. After you're satisfied that SAS has gotten the variable types right, the next questions are:

- *Find unusual observations—are there outliers or incorrect values?*

Use Insight for quick scatterplots, Proc Univariate to identify extreme observations, Proc Freq for list of distinct values

- *What is the pattern of missing data?*

Proc Means will count missing values for a variable

Proc Means `nmiss` data=A;

- *Should some variables be transformed?*

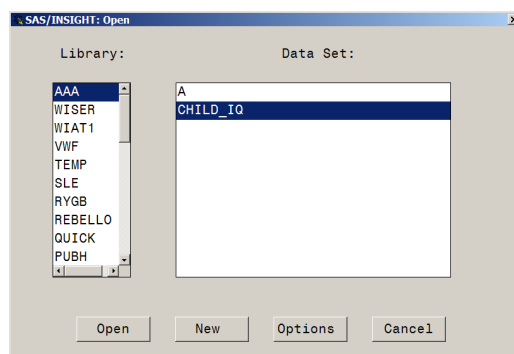
For positive variables, when  $\frac{\text{maximum value}}{\text{minimum value}} > 10$ , take logs.

29

### Quick scatterplots and histograms: Proc Insight

From the menu: Solutions > Analysis > Interactive Data Analysis to start Insight.

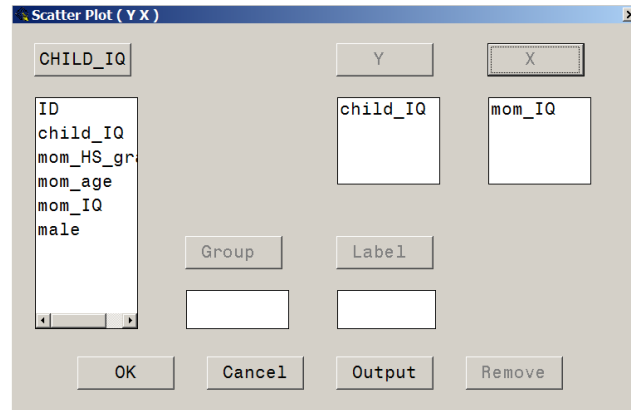
Insight asks you to select a data set from a list of libraries and their SAS datasets.



Select library AAA, then dataset CHILD\_IQ.

30

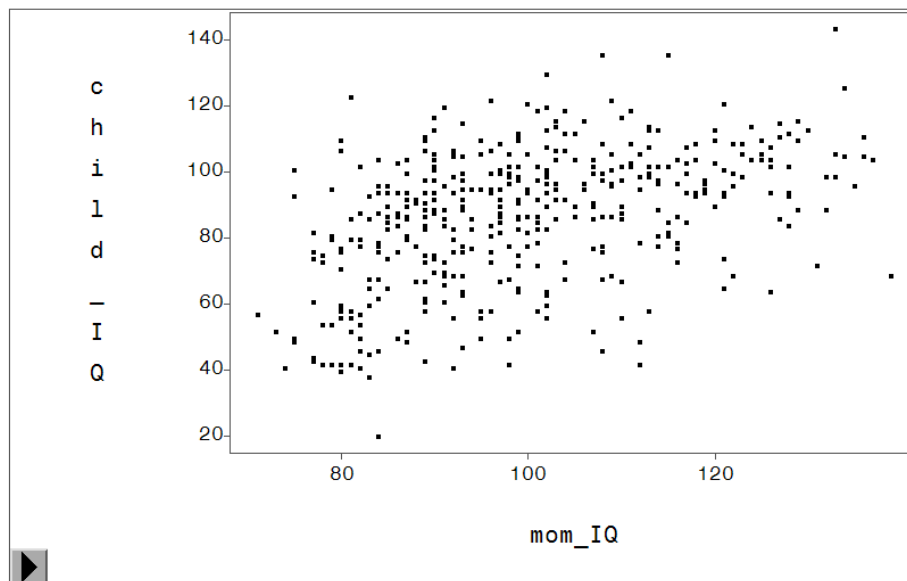
To make a scatterplot, select the menu: Analyze > Scatter Plot (Y X) which brings up a dialog box for selecting the variables to plot:



Select CHILD\_IQ, then click Y. Select MOM\_IQ, then click X, then click OK.

Selecting more than one X or Y gives a matrix of scatter plots.

31



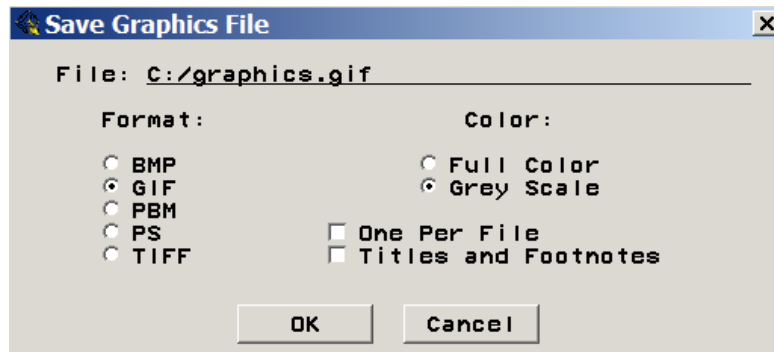
Click on a point to get observation number.

Button on lower left offers different size plotting characters.

Won't give a plot with different symbols for males and females.

32

1. To save an Insight graphic on a PC, copy and paste it into MSWord.
2. On a Mac, use Grab to capture the picture (in TIFF format). Use Preview to convert the file to GIF format, to insert into MSWord.
3. On a PC or a Mac, choose Edit > Save > Graphics File ... which brings up this dialog box:



Type in a path to save to a particular location; default is graphics within the SAS folder in Program Files. Save in GIF format.

33

### **NHANES III.**

The third National Health and Nutrition Examination Survey collected data from 33,994 people during 1988–1994, a representative sample of the whole US population.

We will consider the subset of survey participants aged 20 to 29, which is the data for HW.

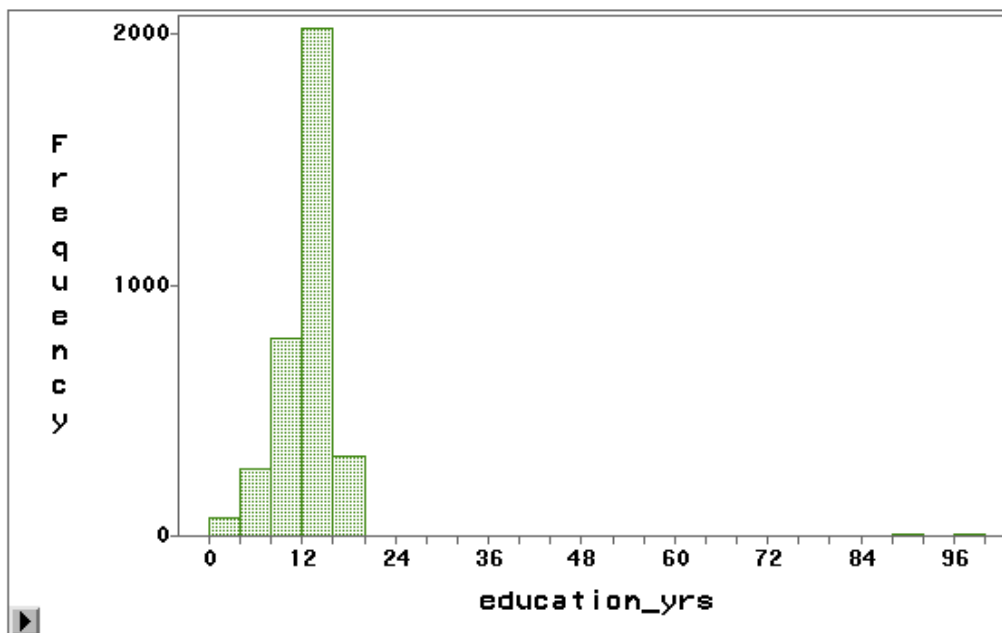
One of the survey questions was: How many years of school have you had?

*What do you expect a histogram of this data to look like?*

34

To make a histogram, choose Analyze > Histogram/Bar Chart (Y) which only makes frequency histograms.

Select education\_yrs, click Y, then click OK.



35

### Proc Univariate

Univariate produces univariate statistics and lists extreme values. The `plot` option gives stem-leaf diagrams, histograms, Q-Q-plots.

```
Proc Univariate plot data=class.nhanes20;
var education_yrs;
```

The UNIVARIATE Procedure  
Variable: education\_yrs

#### Moments

N	3507	Sum Weights	3507
Mean	12.1103507	Sum Observations	42471
Std Deviation	7.31065017	Variance	53.4456059
Skewness	9.14954351	Kurtosis	102.081185
Uncorrected SS	701719	Corrected SS	187380.294
Coeff Variation	60.3669566	Std Error Mean	0.12344915

36

Basic Statistical Measures

Location		Variability	
Mean	12.11035	Std Deviation	7.31065
Median	12.00000	Variance	53.44561
Mode	12.00000	Range	99.00000
		Interquartile Range	3.00000

Quantile	Estimate
100% Max	99
99%	17
95%	16
90%	15
75% Q3	13
50% Median	12
25% Q1	10
10%	8
5%	6
1%	1
0% Min	0

37

List of 5 smallest and 5 largest observations can identify outliers or errors

The UNIVARIATE Procedure  
Variable: education\_yrs

Extreme Observations

----Lowest----		----Highest---	
Value	Obs	Value	Obs
0	3382	99	2309
0	3185	99	2526
0	3141	99	2534
0	2885	99	2771
0	2829	99	3058

*What does 99 years of education mean?*

38



What are these extremely high values of education\_yrs?

Use Proc Freq to list all distinct values.

```
Proc FREQ data=class.nhanes20;  
  tables education_yrs;
```

education_yrs	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	29	0.83	29	0.83
1	8	0.23	37	1.06
2	17	0.48	54	1.54
.	.	.	.	.
11	246	7.01	1139	32.48
12	1268	36.16	2407	68.63
13	297	8.47	2704	77.10
14	290	8.27	2994	85.37
15	169	4.82	3163	90.19
16	219	6.24	3382	96.44
17	102	2.91	3484	99.34
88	11	0.31	3495	99.66
99	12	0.34	3507	100.00

23 observations have nonsense values for years of education.

41

The variable education\_yrs is really HFA8R in the NHANES III data.

From the documentation for NHANES III adult data:

```
HFA8R    What is the highest grade or year of  
         regular school -- has completed?
```

00 Never attended or kindergarten only

01-17

88 Blank but applicable

99 Don't know

88 and 99 are codes for missing data.

```
So exclude values larger than 17 years.
```

42

## Editing data in a DATA step

We want to exclude values of `education_yrs` larger than 17 years.

**Don't edit a spreadsheet. Make the change in the SAS code.**

**This documents the edit.**

```
Data one;
```

```
  SET class.nhanes20;
```

```
  IF (education_yrs LE 17) ; * omit obs with missing education;
```

Better approach—replace 88 and 99 with “missing;” a period indicates a missing number:

```
Data one;
```

```
  SET class.nhanes20;
```

```
  IF (education_yrs > 17) THEN education_yrs = . ;
```

43

Still better—keep the original variable and make a new corrected variable:

```
Data one;
```

```
  SET class.nhanes20;
```

```
  education_yrs_corrected = education_yrs;
```

```
  IF (education_yrs_corrected > 17) THEN education_yrs_corrected = .
```

This doesn't work:

```
Data one;
```

```
  SET class.nhanes20;
```

```
  IF (education_yrs > 17) THEN education_yrs_corrected = . ;
```

`education_yrs_corrected` will be missing for all observations.

Make the variable first, then edit it.

44