Making Changes to Data in SAS
Session 2
(presented by Lizzy Zahn and Lizette Ortega)

OUTLINE

- **Assignment statements**
- **Evaluation of numeric expressions**
- **SAS numeric functions**
- **Logical expressions**
- **IF-THEN statement**
- **Working with character variables**

Assignment statements

The basic method of adding/modifying to a SAS data set is to create or redefine a variable in a DATA step with an ***assignment statement***.

An assignment statement has the form:

variable=expression;

ASSIGNMENT STATEMENTS…

Examples of Assignment statements

Type of expression	Assignment statement
Numeric constant	Yearnow=2009;
Character constant	Progname=’SAS’;
A variable	Newvar=Oldvar;
Addition	Addvalue=Oldvalue+100;
Subtraction	Lessvalue=Oldvalue-100;
Multiplication	Timesvalue=Oldvalue*5;
Division	Splitvalue=Oldvalue/5;
Exponentiation	Powervalue=Oldvalue^5;

Evaluation of numeric expressions

- For numeric expressions with more than one arithmetic operator, SAS follows the standard mathematical rules for order of precedence.

  1) Exponentiation

  2) Multiplication and Division

  3) Addition and Subtraction

- For operators with equal precedence, operations are performed left to right, except for exponentiation, which is performed right to left.

- When parentheses are used, operations in parentheses are performed first.

Evaluation of numeric expressions

By following rules of precedence

X = 4+5*7+4**2;   gives X=55

Adding parentheses,

X = (4+5)*7+4**2;   gives X=79

Missing values in arithmetic expressions

When you use a missing value in an arithmetic expression, SAS sets the result of the expression to missing

Numeric expressions…

Example 1: Tour-example (part1)

```
/* 1) Read tours*/
DATA tours;
INPUT country $ 1-11 group nights aircost  landcost  vendor $;
datalines;
Japan        1  8  982 1020 Express
Greece       2 12    .  748 Express
New Zealand  1 16 1368 1539 Southsea
Ireland      2  7  787  628 Express
Venezuela    2  9  426  505 Mundial
Italy        2  8  852  598 Express
USSR         2 14 1106 1024 A-B-C
Switzerland  2  9  816  924 Tour2000
Australia    1 12 1299 1169 Southsea
Brazil       2  8  682  610 Almeida
;
RUN;
PROC PRINT DATA=tours;
TITLE 'Data Set Tours';
RUN;
```

Numeric expressions…

Example 1: Tour-example (part1)

Data Set Tours

Obs	**country**	**group**	**nights**	**aircost**	**landcost**	**vendor**
1	Japan	1	8	982	1020	Express
2	Greece	2	12	.	748	Express
3	New Zealand	1	16	1368	1539	Southsea
4	Ireland	2	7	787	628	Express
5	Venezuela	2	9	426	505	Mundial
6	Italy	2	8	852	598	Express
7	USSR	2	14	1106	1024	A-B-C
8	Switzerland	2	9	816	924	Tour2000
9	Australia	1	12	1299	1169	Southsea
10	Brazil	2	8	682	610	Almeida

Numeric expressions…

Example 1: Tour-example (part1)

```
/* 2) Create newtours data by modifying tours data */
DATA newtours; SET tours;

/* 3) add aircost and landcost to calculate totalcost */
totalcost=aircost+landcost;

/* 4) calculate peak season land package cost by increasing basic
cost by 20% */
peakland=landcost*1.20;

/* 5) calculate rate per night of basic land package */
pernightland=landcost/nights;
RUN;

PROC PRINT DATA=newtours;
   VAR Country Nights AirCost LandCost TotalCost--Pernightland;
   FORMAT pernightland 7.2;
   TITLE 'Tours Example (Part1) -- Costs for Tours';
RUN;
```

Numeric expressions…

Example 1: Tour-example (part1)

Tours Example (Part1) -- Costs for Tours

Obs	country	nights	aircost	landcost	totalcost	peakland	pernightland
1	Japan	8	982	1020	2002	1224.0	127.50
2	Greece	12	.	748	.	897.6	62.33
3	New Zealand	16	1368	1539	2907	1846.8	96.19
4	Ireland	7	787	628	1415	753.6	89.71
5	Venezuela	9	426	505	931	606.0	56.11
6	Italy	8	852	598	1450	717.6	74.75
7	USSR	14	1106	1024	2130	1228.8	73.14
8	Switzerland	9	816	924	1740	1108.8	102.67
9	Australia	12	1299	1169	2468	1402.8	97.42
10	Brazil	8	682	610	1292	732.0	76.25

Numeric expressions… Now it is your turn!!

Programming problem1: Wages

* Use the “WAGE-program” to create the “wages” data.

* This data file includes the following variables for each ID:

totaldays = total number of days worked

wageday = wage per day (working 8 hours/day)

* From the “wages” data create a new SAS data file “wages2” and create the following new variables:

1) total pay

2) hourly rate

3) monthly rate (assuming 20 work days per month)

* Print the new data file.

Numeric expressions…

Programming problem1: Wages

```
DATA wages;
INPUT id  totaldays wageday;
CARDS;
101      55     165.10
156      35     132.56
...

data wages2;
        set wages;
        totalpay=totaldays*wageday;  /* 1) compute total pay */

        hourlyrate=wageday/8;        /* 2)   hourly rate */

        monthpay=wageday*20;         /* 3) compute wage per month */
run;

PROC PRINT data = wages2;
format hourlyrate 10.2 monthpay 10.2;
TITLE 'Wages data';
run;
```

Numeric expressions…

Programming problem1: Wages

Wages data

id	totaldays	wageday	totalpay	hourlyrate	monthpay
101	55	165.10	9080.50	20.64	3302.00
156	35	132.56	4639.60	16.57	2651.20
204	125	115.89	14486.25	14.49	2317.80
245	78	155.25	12109.50	19.41	3105.00
397	32	112.90	3612.80	14.11	2258.00
456	44	118.21	5201.24	14.78	2364.20
678	67	156.20	10465.40	19.53	3124.00
875	95	134.00	12730.00	16.75	2680.00
941	88	122.45	10775.60	15.31	2449.00

Calculating Numbers Using SAS Functions

A **<u>SAS function</u>** performs a computation or a manipulation of the arguments and returns a value.

SAS functions have the following general form

function-name(argument, argument, …)

SAS has around 280 built-in numeric functions

Selected SAS Numeric Functions

Function	Definition	Example	Result

N	Number of non-missing values	X1=n(1,3,2);
		X2=n(2, . ,5);
			3
			2

SUM	Sum of the non-missing arguments	S1=sum(2,3,5);
		S2=sum(1, . , 3, 4);
		S3=sum(S1,S2);
			10
			8
			18

MEAN	Arithmetic mean (average)	M1=mean(S1,S2);
		M2=mean(of S1-S3);
			9
			12

MIN	Smallest value	Y=min(X1,of S1-S3,X2);	2
MAX	Largest value	Z=max(X1,of S1-S3,X2);	18
INT	Integer portion of the argument	XI=int(10.25);	10
ROUND	Rounds the first argument to the nearest multiple of the second argument, or to the nearest integer when the second argument is omitted	R=round(106,10);	110
LOG10	Base 10 logarithm	SLOG=LOG(S1);	1

Numeric expressions…

Example 1: Tour-example (part2)

```
/* 6) Create roundtours data from tours data */
DATA roundtours; SET tours;

/* 7) Round aircost to nearest 50 */
roundair=round(aircost,50);

/* 8) Round totalcost to nearest 100 */
totalcostr=round(aircost+landcost,100);

/* 9) Calculate total cost based on all non-missing costs */
costsum=sum(aircost,landcost);

/* 10) Round total cost based on all non-missing costs to the
nearest 100 */
roundsum=round(costsum,100);
RUN;

PROC PRINT DATA= roundtours;
TITLE 'Tours Example (Part2) -- rounding and summing variables';
RUN;
```

Numeric functions…

Example 1: Tour-example (part2)

```
Tours Example (Part2) -- rounding and summing variables

country       group   nights   aircost   landcost   vendor     roundair   totalcostr   costsum   roundsum

Japan           1        8        982      1020     Express      1000        2000        2002      2000
Greece          2       12          .       748     Express         .           .         748       700
New Zealand     1       16       1368      1539     Southsea     1350        2900        2907      2900
Ireland         2        7        787       628     Express       800        1400        1415      1400
Venezuela       2        9        426       505     Mundial       450         900         931       900
Italy           2        8        852       598     Express       850        1500        1450      1500
USSR            2       14       1106      1024     A-B-C        1100        2100        2130      2100
Switzerland     2        9        816       924     Tour2000      800        1700        1740      1700
Australia       1       12       1299      1169     Southsea     1300        2500        2468      2500
Brazil          2        8        682       610     Almeida       700        1300        1292      1300
```

Comparing Numeric Variables: IF/THEN Statements

Logical operators used for variable comparisons

Logical operation	Symbol/Name equivalent
Equal	= / EQ
Not equal	^= / NE
Greater than	> / GT
Greater than or equal	>= / GE
Less than	< / LT
Less than or equal	<= / LE

Logical operators…

Example 1: Tour-example (part3)

11) Create “newtours2” data by modifying “newtours” data

12) Calculate peak air cost which is greater than basic cost
    by 15% with extra $20 fee for group=1 tours and
    by 10% with extra $15 fee for group=2

13) Calculate peak season total cost

14) Classify cost as 'HIGH' if totalcost is $2000 or more OR per night land cost $100 or more.
    Otherwise, classify as 'OK' if totalcost is <$2000 and per night cost is <$100.
    If not HIGH and either totalcost or per night cost is missing, not able to classify.

Logical operators IF-THEN…

Example 1: Tour-example (part3)

```
 /* 11) Create newtours2 data from newtours data */
DATA newtours2;
        SET newtours;

/* 12) Calculate peak air cost which higher than basic cost by 15%
with extra $20  fee for group=1 tours*/
        IF group=1 THEN peakair=aircost*1.15+20;

 /* and by 10% with extra $15 fee for group=2 tours*/
        ELSE IF group=2 THEN peakair=aircost*1.10+15;

 /* 13) calculate peak season total cost */
        peaktotal=peakair+peakland;

/* 14) Classify cost as 'HIGH' if totalcost is >$2000 OR per night
land cost >$100. */
```

Logical operators IF-THEN…

Example 1: Tour-example (part3 cont…)

```
/* 14) Classify cost as 'HIGH' if totalcost is >$2000 OR per night
land cost >$100. */
       IF totalcost>=2000 OR pernightland>=100 THEN
       costclass='HIGH';

/*Otherwise, classify as 'OK' if totalcost is <$2000 and per night
cost is <$100.
if not HIGH and either totalcost or per night cost is missing, not
able to classify */
       ELSE IF .<totalcost<2000 AND .<pernightland<100 THEN
       costclass='OK';
RUN;

PROC PRINT DATA =newtours2;
FORMAT pernightland 7.2;
TITLE 'Tours Example (Part3) -- Use of logical operators in IF-THEN
statement';
RUN;
```

Logical operators IF-THEN…

Example 1: Tour-example (part3)

```
country    group nights aircost landcost vendor   totalcost peakland pernightland peakair peaktotal costclass

Japan         1      8      982    1020   Express     2002    1224.0     127.50    1149.30  2373.30    HIGH
Greece        2     12        .     748   Express        .     897.6      62.33        .        .
New Zealand   1     16     1368    1539   Southsea    2907    1846.8      96.19    1593.20  3440.00    HIGH
Ireland       2      7      787     628   Express     1415     753.6      89.71     880.70  1634.30    OK
Venezuela     2      9      426     505   Mundial      931     606.0      56.11     483.60  1089.60    OK
Italy         2      8      852     598   Express     1450     717.6      74.75     952.20  1669.80    OK
USSR          2     14     1106    1024   A-B-C       2130    1228.8      73.14    1231.60  2460.40    HIGH
Switzerland   2      9      816     924   Tour2000    1740    1108.8     102.67     912.60  2021.40    HIGH
Australia     1     12     1299    1169   Southsea    2468    1402.8      97.42    1513.85  2916.65    HIGH
Brazil        2      8      682     610   Almeida     1292     732.0      76.25     765.20  1497.20    OK
```

Working with Character Variables

A **character variable** is a variable whose value contains letters, numbers, and special characters, and whose length can be from 1 to 32,767 characters long

Character variables can be used in

- declarative statements
- comparison statements
- assignment statements

Selected SAS Character Functions

Function	Definition	Syntax
INDEX	Returns starting position for string of characters	INDEX(arg,’string’)
SUBSTR	Extracts a substring from an argument starting at position for n characters or until end if no n	SUBSTR(arg,position,n)
TRIM	Removes trailing blanks from character expression	TRIM(arg);
UPCASE	Converts all letters in argument to uppercase	UPCASE(arg)
TRANWRD	Replaces “from” character string in source with “to” character string	TRANWRD(source,from,to)
LENGTHN	Returns the number pf non-blank characters of an argument	LENGTHN(arg)

Working with Character Variables…

Example 2: Words-example

```
/* 1) Read "Words" data */
DATA wordsonly;
       SET words;

/* 2) Use IF-THEN statement to delete non-words in data file */
       IF word<'A' THEN DELETE;

/* 3) Use SUBSTR function to create a character variable that
       contains the first letter of the word syntax is
       SUBSTR(variable,start,number-characters) */
       StartWord=SUBSTR(word,1,1);

/* Use the LENGTHN function to create a variable that contains
the number of non-blank characters of each word */
       nletters=LENGTHN(word);
RUN;
PROC PRINT data = wordsonly;
ID WORD;
run;
```

Working with Character Variables…

Example 2: Words-example

```
LINCOLN'S GETTYSBURG ADDRESS
frequencies of starting letter and word lengths

               Start
WORD           Word     nletters

FOURSCORE        F          9
AND              A          3
SEVEN            S          5
YEARS            Y          5
AGO              A          3
OUR              O          3
FATHERS          F          7
BROUGHT          B          7
FORTH            F          5
ON               O          2
THIS             T          4
CONTINENT        C          9
A                A          1
NEW              N          3
NATION           N          6
CONCEIVED        C          9
...
```

Working with Character Variables…

Example 2: Words-example

```
PROC FREQ;
TABLES startword nletters;
TITLE “LINCOLN’S GETTYSBURG ADDRESS”;
TITLE2 “frequencies of starting letter and word lengths”;
RUN;
```

Working with Character Variables…

Example 2: Words-example

```
LINCOLN'S GETTYSBURG ADDRESS
frequencies of starting letter and word lengths
Start                             Cumulative    Cumulative
Word     Frequency     Percent     Frequency      Percent
-----------------------------------------------------------
A              24        8.99            24         8.99
B              10        3.75            34        12.73
C              14        5.24            48        17.98
D              15        5.62            63        23.60
E               4        1.50            67        25.09
F              19        7.12            86        32.21
G               8        3.00            94        35.21
H              16        5.99           110        41.20
I              13        4.87           123        46.07
L              10        3.75           133        49.81
M               5        1.87           138        51.69
N              14        5.24           152        56.93
O              11        4.12           163        61.05
P              10        3.75           173        64.79
R               6        2.25           179        67.04
S              11        4.12           190        71.16
T              47       17.60           237        88.76
U               5        1.87           242        90.64
V               1        0.37           243        91.01
W              23        8.61           266        99.63
Y               1        0.37           267       100.00
```

Working with Character Variables…

Example 2: Words-example

```
LINCOLN'S GETTYSBURG ADDRESS
frequencies of starting letter and word lengths
                                     Cumulative    Cumulative
nletters    Frequency     Percent     Frequency      Percent
-------------------------------------------------------------
       1           7        2.62             7         2.62
       2          50       18.73            57        21.35
       3          54       20.22           111        41.57
       4          57       21.35           168        62.92
       5          33       12.36           201        75.28
       6          27       10.11           228        85.39
       7          15        5.62           243        91.01
       8           6        2.25           249        93.26
       9          11        4.12           260        97.38
      10           4        1.50           264        98.88
      11           3        1.12           267       100.00
```

Now it is your turn again!!

Programming Problem 2: Grades – Calculate Final Grade

Use the “GRADES-program” to create the “exam” data.

This data file includes the following variables for each ID:

4 quiz scores = QUIZ1, QUIZ2, QUIZ3, QUIZ4

final exam grade = final

Create finalgrade data from exam data that includes the following new variables:

1) The variable NQUIZ that contains the number of quizzes taken (use N function)

2) The variable TOTALQUIZ that has the total points for quizzes (SUM function)

3) If the student took all 4 quizzes, recalculate TOTALQUIZ by subtracting lowest quiz score (use MIN function)

4) Compute average quiz score (AVEQUIZ), by dividing TOTALQUIZ by 3

5) Compute final grade (FINALGRADE), which is 60% of average quiz score and 40% of final exam

6) USE IF-THEN and ELSE IF statements to assign letter grades (GRADE) based on the final grade, where 90 or higher is A, 80 to less than 90 is B, 70 to less than 80 is C, 60 to less than 70 is D, and below 60 is F.

Print the new data file.

Programming Problem 2: Grades – Calculate Final Grade

```
data finalgrade;
      set exam;

/* 1) Number of quizzes taken */
      nquiz=N(OF quiz1-quiz4);

/* 2) total points for quizzes */
      totalquiz=SUM(OF quiz1-quiz4);

/* 3) subtract lowest quiz score from totalquiz if 4 quizzes
taken */
      IF nquiz=4 THEN totalquiz=totalquiz-MIN(OF quiz1-
      quiz4);

/* 4) total quiz score divided by 3 */
      avequiz=totalquiz/3;

/* 5) final grade is 60% average quiz score and 40% of final
exam */
      finalgrade=0.60*avequiz+0.40*final;
```

Programming Problem 2: Grades – Calculate Final Grade

```
/* 6) assign letter grade */
       IF finalgrade>=90 THEN grade=’A’;
       ELSE IF finalgrade>=80 THEN grade= ’B’;
       ELSE IF finalgrade>=70 THEN grade=’C’;
       ELSE IF finalgrade>=60 THEN grade=’D’;
       ELSE IF .<finalgrade<60 THEN grade=’F’;
RUN;

PROC PRINT DATA=finalgrade;
ID idno;
TITLE ’FINAL GRADES’;
RUN;
```

Programming Problem 2: Grades – Calculate Final Grade

```
FINAL GRADES

idno   quiz1   quiz2   quiz3   quiz4   final   nquiz   totalquiz   avequiz   finalgrade   grade

 101     80      75      90      85      88      4        255      85.0000      86.2        B
 102     90      75      80      82      72      4        252      84.0000      79.2        C
 103     80      55      83      75      66      4        238      79.3333      74.0        C
 104     77      65      79      68      70      4        224      74.6667      72.8        C
 105     86      90      77      92      91      4        268      89.3333      90.0        A
 106     82      78      69      65      62      4        229      76.3333      70.6        C
 107     90      89      86       .      92      3        265      88.3333      89.8        B
 108     66       .      55       .      51      2        121      40.3333      44.6        F
 109      .      65      74      78      75      3        217      72.3333      73.4        C
 110     52      75      81      86      89      4        242      80.6667      84.0        B
 111     71      68      73      77      83      4        221      73.6667      77.4        C
 112     89      91      92      70      90      4        272      90.6667      90.4        A
 113     60      58      65      45      65      4        183      61.0000      62.6        D
 114     81      75      85      90      95      4        256      85.3333      89.2        B
 115     95      94      91       .      93      3        280      93.3333      93.2        A
```