STAT 2005 – Programming Languages for Statistics

Tutorial 9 Data Manipulation

2020

LIU Ran

Department of Statistics, The Chinese University of Hong Kong

1 Usage of Equal Sign

1.1 Assignment Statement

If itself is a statement as we have a semicolon, we will treat the equal signs as comparison except for the first equal sign. The first equal sign will be treated as assignment. Because a statement must do something, it cannot just be an expression.

```
Data;
A = 1;B = 2;C=1;
D = A = B = C;
E = A = C;
Run;
```

Because B is not equal to A or C, the `A=B=C` will return zero. And SAS will assign zero to the variable D.

Because A is equal to C, the `A=C` will return one. And SAS will assign one to the variable E.

1.2 Expression

If it is just an expression, `B = A = C` will return one if B is equal to A and C, otherwise zero.

```
Data;
A = 1;B = 2; C=1;
IF B = A = C then D = 1;
Run;
```

Remark 1.1. When using comparison operators, it is better to use the alternative symbol `EQ` instead of =.

	Symbol	Alt. symbol	Meaning
Comparison Operators :	=	EQ	Equal, e.g. a = b , a eq b (gives value TRUE if and only if a = b)
	^=	NE	Not equal, e.g. a ^= b, a ne b (gives value TRUE if and only if a is not equal to b)
	>	GT	Greater than, e.g. a > b, a gt b
	>=	GE	Greater than or equal, e.g. a >= b, a ge b
	<	LT	Less than, e.g. a < b, a lt b
	<=	LE	Less than or equal, e.g. a <= b, a le b
Logical (Boolean) operators :	^	NOT	(prefix) negation, e.g. ^(a+b = 4)
	&	AND	And, e.g. a = 1 & b = 2 , a = 1 and b = 2
	|	OR	Or, e.g. a = 1 | b = 2, a = 1 or b = 2
Other operator :		IN	List membership, e.g. a in (6, 7, 8) (it gives TRUE value if and only if a is 6, 7 or 8), region in ('NE', 'W', 'S')

2 The Length of Variables

When re-assign value to the variable, the length will not change.

```
Data;
A = 'Mr '; B = 'Liu';
C = A||B;
A = A||B;
Run;
```

it will return the result

```
    A     B     C
1   Mr  Liu   Mr Liu
```

It is because the length of the variable A has already been set to be three, 'Mr '. If we want to achieve the assignment, as you have learned in class, we should define the length in advanced.

```
Data;
LENGTH A $ 10;
A = 'Mr '; B = 'Liu';
A = A||B;
Run;
```

However, if you try the above command, you will find `A` is again equal to `'Mr '`. What happened? If you look at the slide in class carefully, the example uses the target variable behind the operator `||`. So, we try to put A behind the operator:

```
Data;
LENGTH A $ 10;
A = 'Mr '; B = 'Liu';
A = B||A;
Run;
```

It works! we will have the result `'LiuMr'`. Why?

I checked some reference and found that it is because when you define `LENGTH A $ 10;A = 'Mr ';`. Actually, A will be 'Mr', the two letters, and eight blanks behind the letters. So, when you try to concatenate A and B, `A||B`, the result will be 'Mr' and eight blanks and 'Liu'. And the length of A is 10, so the last three letters 'Liu' is truncated and removed, only remains 'Mr' and eight blanks.

If you use `B||A`, the last three blanks in A are truncated, so the result is 'LiuMr' and some blanks.

To solve this problem, try to use the command `strip` to remove the blanks behind A.

```
Data;
LENGTH A $ 10;
A = 'Mr '; B = 'Liu';
A = strip(A)||B;
Run;
```

The result is `'MrLiu'`.

Remark 2.1. reference

3 ROUND

Unlike the usage in R, it will round the first argument to the nearest multiple of the second argument, or to the nearest integer when the second argument is omitted.

`round(3.23, 0.1) = 3.2, round(2.23) = 2, round(8.613,0.5) = 8.5.`

4 Random number generation

RAN + Distribution name: `RANBIN(seed,n,p), RANNOR(seed), RANUNI(seed)`

Unlike R, we should set seed whenever generate the data. And if we set seed to 0 or a negative integer, the system time will be used as the seed. Therefore, we will get different sequence of random numbers in different runs of the program if we set seed to 0.

```
Data;
seed = 0;
n = 10;
p = 0.6;
a = RANBIN(seed,n,p);
run;
```

Each time will give new `a`.

5 Character functions

5.1 LENGTH

The position of the rightmost non-blank character in the string. It is not the length of non-blank characters. It is the position.

`length(' a bcd ')` gives 6, the position of character 'd'.

5.2 INDEX

Search for the first occurrence of a 'word' anywhere in a string. If the string is not found, the result is zero. (must match)

5.3 INDEXC

Locates the first occurrence of any character specified in the excerpt(a short extract). If no target is found, the result is zero.

5.4 INDEXW

Finds the target excerpt in a string on a word boundary. If the word is not found, the result is zero.

The substring pattern must begin and end on a word boundary. For `INDEXW`, word boundaries are blanks, the beginning of source, and the end of source. Punctuation marks are not word boundaries.

```
Data;
password = 'david999777 work do';
a = INDEX(password, 'david');
b = INDEX(password, 'david1');
c = INDEXC(password, '0123456789');
d = INDEXW(password, 'work');
e = INDEXW(password, 'david');
run;
```

result: `a=1, b=0, c=6, d=13, e=0`.

Remark 5.1. reference

6 Conditional execution: IF statement

```
IF condition THEN action1; [ELSE action2;]
```

Remark 6.1. There should be no semicolon between IF and THEN; and should be a semicolon between THEN and ELSE.

Remark 6.2. Unlike R, SAS don't have the double and sign && and the `||` has different meaning, character concatenation.

7 Variable List

A SAS variable list is an abbreviated method of referring to a list of variable names. SAS enables you to use the following variable lists:

1. numbered range lists
2. name range lists
3. name prefix lists
4. special SAS name lists

7.1 Numbered Range Lists

Numbered range lists require you to have a series of variables with the same name, except for the last character or characters, which are consecutive numbers. For example, the following two lists refer to the same variables:

```
Var1 Var2 Var3 Var4 Var5 Var6

Var1-Var6
```

For example

```
data exam;
   input Score11-Score20;
datalines;
1 2 3 4 5 6 7 8 9 10
;
```

7.2 Name Range Lists

Name range lists rely on the order of variable definition, as shown in the following:

1. `vara -- varb`: all variables in order of variable definition, from variable a to variable b inclusive

```
data patients;
   input Idnum Name $ Weight Pulse BMI Gender $;
datalines;
123 Jones 155 82 27 F
456 Smith 175 78 24 M
789 Kamda 172 69 22 F
;

data patientsConsec;
   set patients;
   keep Name--Pulse;
run;
proc print data=patientsConsec; run;
```

2. `vara -NUMERIC- varb`: all numeric variables from variable a to variable b inclusive

```
data patientsNumerics;
   set patients;
   keep Idnum-numeric-BMI;
run;
proc print data=patientsNumerics; run;
```

3. `vara -CHARACTER- varb`: all character variables from variable a to variable b inclusive

```
data patientsCharacters;
   set patients;
   keep Idnum-character-Pulse;
run;
proc print data=patientsCharacters; run;
```

7.3 Name Prefix Lists

Some SAS functions and statements enable you to use a name prefix list to refer to all variables that begin with a specified character string:

```
sum(of Sales:)
```

7.4 Special SAS Name Lists

1. `_CHARACTER_`: specifies all character variables that are already defined in the current DATA step.
2. `_NUMERIC_`: specifies all numeric variables that are already defined in the current DATA step.
3. `_ALL_`: specifies all variables that are already defined in the current DATA step.

Remark 7.1. reference

8 Exercises

1. If variables in the data set are ordered as the following:

   ```
   x1 x2 x3 Sex $ Grade $ x5 income x4
   ```

   Write down the resulting variable list:

   (a) `x1 - x5`
   (b) `x1 -- x5`
   (c) `x1 -NUMERIC- Sex`
   (d) `x2 -CHARACTER- Grade`
   (e) `_CHARACTER_`
   (f) `_NUMERIC_`

   (a) x1 x2 x3 x4 x5 (b) x1 x2 x3 Sex Grade x5 (c) x1 x2 x3 (d) Sex Grade (e) Sex Grade (f) x1 x2 x3 x5 income x4

9 Character comparison

not('ab '='a'——"b") will result in 0. Two character values of unequal length are compared as if blanks were attached to the end of the shorter value before the comparison is made. A blank, or missing character value, is smaller than any other printable character value.

Remark 9.1. reference

10 How to upload SAS file and SAS code in website version

First, create a new library named test: click 'New Library' icon, let the path be your home path.

Second, upload the SAS file under the home path.

Third, use the set command to retrieve your data. Data A2; set test.blood; run; where TEST is the library name you created, blood is the SAS file name.