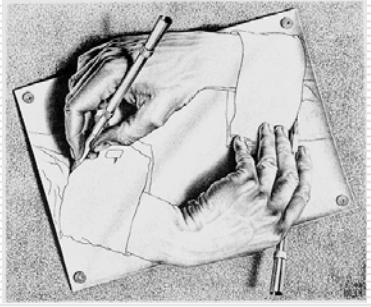


SCSUG

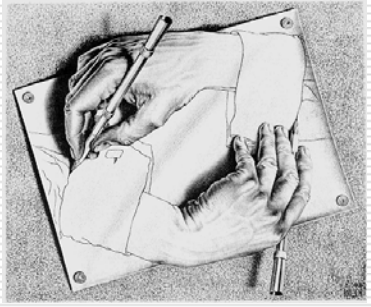
Avoiding Hand Cramps: Name2 macro and arrays for related variables

November 9, 2010 9am

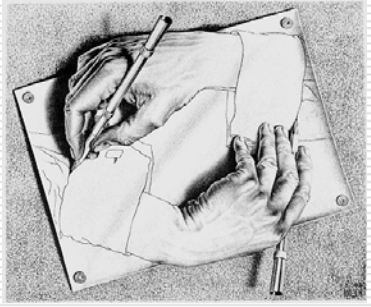
South Center SAS Users Group

Austin, Texas

Problem: Report percentages and per animal-type on subtotal based on family subtotals and zoo census



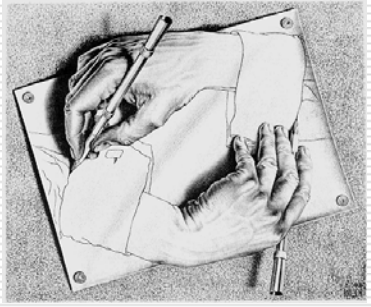
	1	2	3	4	5	6
	FOOD FUND	%	PER ANIMAL	ALL FUNDS	%	PER ANIMAL
Total Operating	EOPTOT1	EOPTOT2	EOPTOT3	EOPTOT4	EOPTOT5	EOPTOT6
TOTAL CANINES	CANINE1	CANINE2	CANINE3	CANINE4	CANINE5	CANINE6
Dogs	DOG1	DOG2	DOG3	DOG4	DOG5	DOG6
Wolves	WOLF1	WOLF2	RLOC3	RLOC4	RLOC5	RLOC6
Foxes	FOX1	FOX2	RSTA3	RSTA4	RSTA5	RSTA6
TOTAL FELINES	FELINE1	FELINE2	ELINE3	ELINE4	FELINE5	FELINE6
Cats	CAT1	CAT2	CAT3	CAT4	CAT5	CAT6
Lions	LION1	LION2	LION3	LION4	LION5	LION6
Tigers	TIGER1	TIGER2	IGER3	IGER4	TIGER5	TIGER6
TOTAL PORCINES	PORCINE	PROCINE	ORCINE	PROCINE	PORCINE	PROCINE
Pigs	PIG1	PIG2	IG3	IG4	PIG5	PIG6
Swine	SWINE1	SWINE2	WINE3	WINE4	SWINE5	SWINE6
Boars	BOAR1	BOAR2	OAR3	OAR4	BOAR5	BOAR6



Macro Code

```
%macro name2(macrovar,suf);
%let i=0;
%do %while(%scan(&macrovar,&i+1,%str( ))
    ne %str( ));
%let i = %eval(&i+1);
%let var=%scan(&macrovar,&i,%str( ));
&var.&suf
%end;
%mend name2;
```

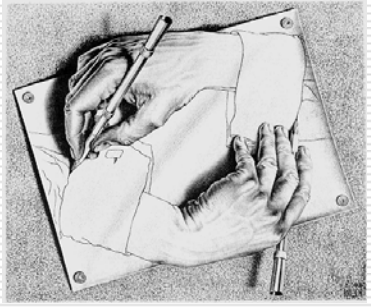
Source: Al McKenzie, Texas Education Agency



%do %while

- can be any macro expression that resolves to a logical value.
- The macro processor evaluates the expression at the top of each iteration.
- The expression is true if it is an integer other than zero.
- The expression is false if it has a value of zero.
- If the expression resolves to a null value or to a value containing nonnumeric characters, the macro processor issues an error message.

Source: http://support.sas.com/onlinedoc/913/docMainpage.jsp?ETS=4431&PID=135574

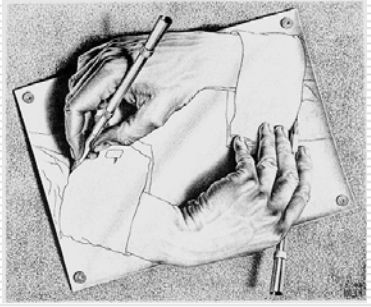


(%scan

- The %SCAN and %QSCAN functions search *argument* and return the *n*th word.
- A word is one or more characters separated by one or more delimiters.
- %SCAN does not mask special characters or mnemonic operators in its result, even when the argument was previously masked by a macro quoting function.
- %QSCAN masks the following special characters and mnemonic operators in its result:

& % ' " ( ) + - * / < > = ¬ ^ ~ ; , blank AND OR NOT EQ NE LE LT GE GT

Source: http://support.sas.com/onlinedoc/913/docMainpage.jsp?ETS=4431&PID=135574



(&macrovar,&i+1,%str(

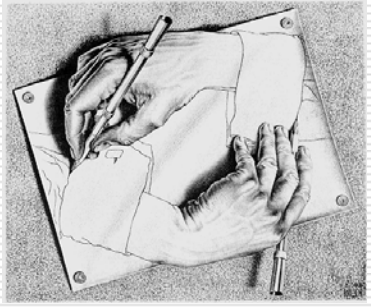
- What is a next value from the parameter list?

  This increments that value and prepares the parameter for additional manipulation

  1. The variable name is retried as the first word.
  2. The %(is used to mark the open parenthesis as a word delimiter
  3. Notice the use of %(and %) to designate the open and close parenthesis.
  4. The list of parameters is temporarily stored in &macrovar
  5. The resulting list of comparisons is passed back

Source: Carpenter's Complete Guide to the SAS Macro Language, Second Edition

ne %str( ));



- If a special character or mnemonic affects the way the macro processor constructs macro program statements, you must mask the item during macro compilation
- (or during the compilation of a macro program statement in open code) by using either the %STR or %NRSTR macro quoting functions.
- These macro quoting functions mask the following special characters and mnemonics:

Blank	)	=	NE
;	(	|	LE
¬	+	#	LT
^	--	AND	GE
~	*	OR	GT
, (comma)		/	NOT
‘	<	IN	
“	>	EQ	

Source: http://support.sas.com/onlinedoc/913/docMainpage.jsp?ETS=4431&PID=135574

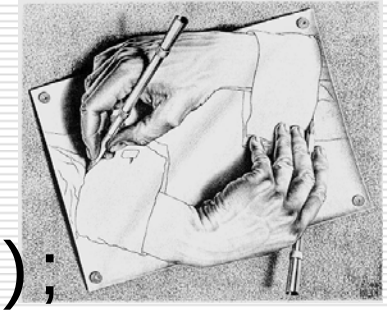
%eval(&i+1);



- The %EVAL function evaluates integer arithmetic or logical expressions. %EVAL operates by converting its argument from a character value to a numeric or logical expression. Then, it performs the evaluation.
- Finally, %EVAL converts the result back to a character value and returns that value.

- If all operands can be interpreted as integers, the expression is treated as arithmetic.
- If at least one operand cannot be interpreted as numeric, the expression is treated as logical.
- If a division operation results in a fraction, the fraction is truncated to an integer. So how does it process?
  1. Checks to see if there is a next word. If there is then enter the loop so that &i can be incremented ï.
  2. Since the &i+1 word was found, increment the value of &i.
  3. When the last word has been counted, exit the loop and return the number of words counted by passing it back.

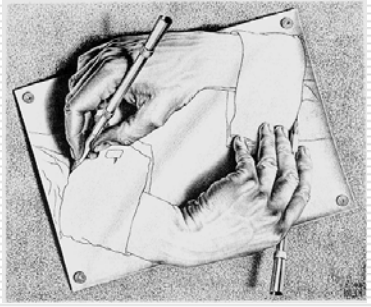
Source: Art Carpenter, http://www2.sas.com/proceedings/sugi30/028-30.pdf



%let var=%scan(&macrovar,&i,%str( ));

- The VARNAME function returns the name of the &Ith variable. This variable is appended to the growing list of variables stored in &VARLIST.
- The %SCAN function can then be used to identify and breakout the individual array elements.
- the %SCAN function have two required and one optional argument and they take the form of:
  - %scan(*macro_var_list, word_number, word_delimiter*)
- %SCAN looks ahead to see if the next word (&COUNT + 1) exists.

Source: Art Carpenter, http://www2.sas.com/proceedings/sugi30/028-30.pdf .

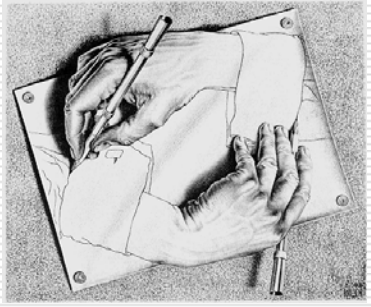


&var.&suf

- **&var**
  - NVAR variable from a given dataset
  - which are contained in a call macro like %tokeep
  - which contains the names of variables of the dataset and create macro variables.
  - Then resolve these macro variables in a macro DO LOOP to perform any necessary function.
- **&suf**
  - add the suffix "_base" to all the variables in the sas dataset.

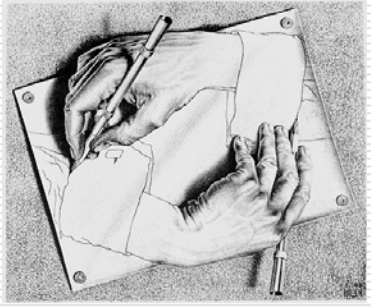
Sources: http://www2.sas.com/proceedings/sugi28/118-28.pdf ,

A second look



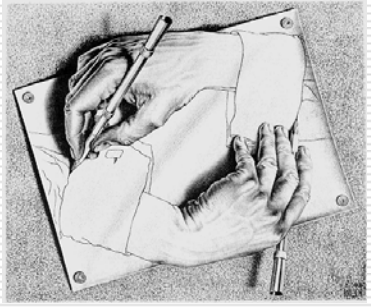
```
%macro name2(macrovar,suf);
%let i=0;
%do
   %while(%scan(&macrovar,&i+1,%str( ))
   ne %str( ));
%let i = %eval(&i+1);
%let var=%scan(&macrovar,&i,%str( ));
&var.&suf
%end;
%mend name2;
   %let tokeep=dog wolf fox canine
              Cat lion tiger feline
              Pig swine boar porcine;
```

Problem: Report percentages and per animal-type on subtotal based on family subtotals and zoo census



	1	2	3	4	5	6
	FOOD FUND	%	PER ANIMAL	ALL FUNDS	%	PER ANIMAL
Total Operating	EOPTOT1	EOPTOT2	EOPTOT3	EOPTOT4	EOPTOT5	EOPTOT6
TOTAL CANINES	CANINE1	CANINE2	CANINE3	CANINE4	CANINE5	CANINE6
Dogs	DOG1	DOG2	DOG3	DOG4	DOG5	DOG6
Wolves	WOLF1	WOLF2	RLOC3	RLOC4	RLOC5	RLOC6
Foxes	FOX1	FOX2	RSTA3	RSTA4	RSTA5	RSTA6
TOTAL FELINES	FELINE1	FELINE2	ELINE3	ELINE4	FELINE5	FELINE6
Cats	CAT1	CAT2	CAT3	CAT4	CAT5	CAT6
Lions	LION1	LION2	LION3	LION4	LION5	LION6
Tigers	TIGER1	TIGER2	IGER3	IGER4	TIGER5	TIGER6
TOTAL PORCINES	PORCINE	PROCINE	ORCINE	PROCINE	PORCINE	PROCINE
Pigs	PIG1	PIG2	IG3	IG4	PIG5	PIG6
Swine	SWINE1	SWINE2	WINE3	WINE4	SWINE5	SWINE6
Boars	BOAR1	BOAR2	OAR3	OAR4	BOAR5	BOAR6

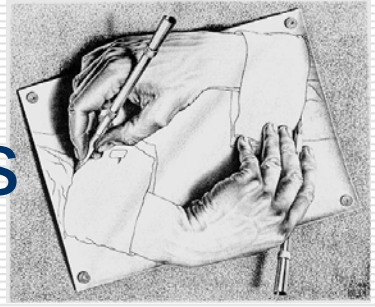
Adding a subtotal to an all total



* Keep in mind that food expense is a subset of all expenses, and other expense types need the addition of food;

```
%let tokeep=DOG WOLF FOX CANINE CAT
   LION TIGER FELINE PIG SWINE BOAR
   PORCINE;
      array items {*} &tokeep ;
      array all {*} %name2(&tokeep,4);
      array food {*} %name2(&tokeep,1) ;
            do i = 1 to dim(items);
                  all{i} =  sum(0, food{i},
   all{i});
            end;
EOBTOT1  = SUM(0, DOG, WOLF, FOX, CAT,
   LION, TIGER, PIG, SWINE, BOAR);
```

Treating for unique denominators

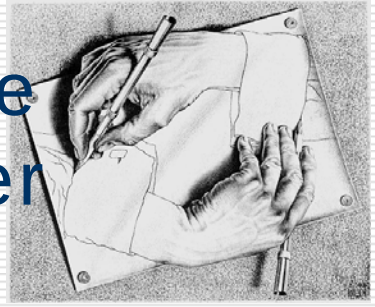


We need to assign a total variable for food totals and another total variable for food and the sum of other totals.

We use the same logic of giving food a suffix of 1 and food and all other expenses sum with a suffix of 4

These are represented as EOBTOT1 AND EOBTOT4 for all of the animals

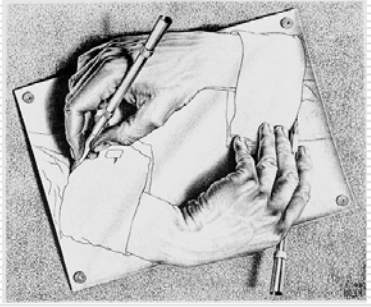
Treating for a global zoo population; use several %tokeep,n, but have one master list



```
%let tokeep1=CANINES DOG WOLF FOX;
%let total1    =CANINES;
%let tokeep2=FELINES CAT LION TIGER;
%let total2    =FELINES;
%let tokeep3=PORCINES PIG SWINE BOAR;
%let total3    =PORCINES;

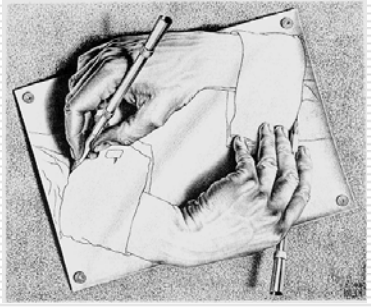
DATA BUDGET(keep=year quarter zoo_pop
%name2(&tokeep,1) %name2(&tokeep,2)
%name2(&tokeep,3)  %name2(&tokeep,4)
%name2(&tokeep,5)  %name2(&tokeep,6));
```

Calculating the sub-expense & total expense for groups, subgroups



```
array allpop  {*} %name2(&tokeep,6);
 array foodpop  {*} %name2(&tokeep,3) ;
 array items {*} &tokeep ;
 array all {*} %name2(&tokeep,4);
 array gen {*} %name2(&tokeep,1) ;
        do i = 1 to dim(items);
     if zoo_pop ne 0 then foodpop{i} = ROUND( gen {i} /
zoo_pop);
        if zoo_pop ne 0 then    allpop{i} = ROUND( all {i} /
zoo_pop);
        end;
```

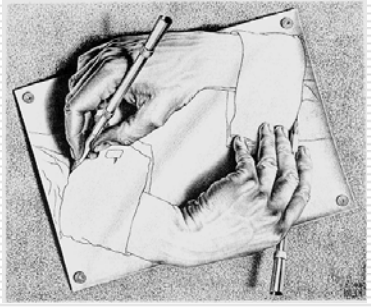
Review: use several %tokeeps with several sub-populations to report



```
%let tokeep1=CANINES DOG WOLF FOX;
%let total1    =CANINES;
%let tokeep2=FELINES CAT LION TIGER;
%let total2    =FELINES;
%let tokeep3=PORCINES PIG SWINE BOAR;
%let total3    =PORCINES;
```

Are different assignments from

```
%let tokeep=DOG WOLF FOX CAT LION TIGER PIG SWINE BOAR;
```

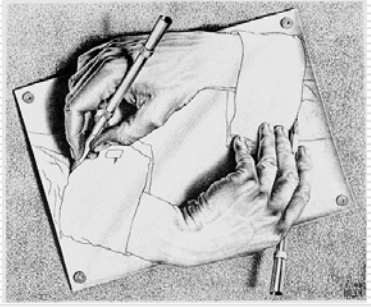


%LET STATEMENT

%let tokeep=DOG WOLF FOX CAT LION TIGER PIG SWINE BOAR;

"%Tokeep" is effective if there is a treatment that requires the total population to apply as a denominator against the master array.

When Denominators differ for different sub-group calculations, count on the numerically suffixed %tokeeps



%let tokeep1=DOG WOLF FOX ;

%let total1 =CANINE;

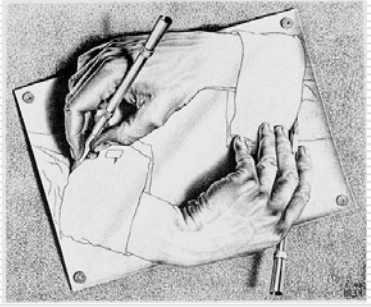
%let tokeep2=CAT LION TIGER;

%let total2 =FELINE;

%let tokeep3=PIG SWINE BOAR;

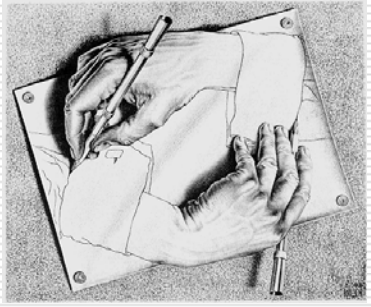
%**let total3 =PORCINE;**

Assigned subtotals act as denominators: What a dog?!



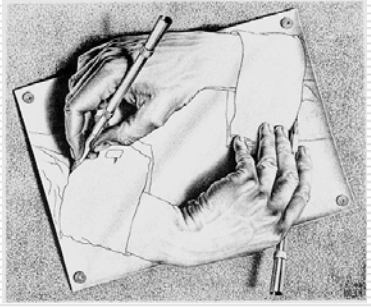
```
*DOG;     array   items1 {*} &tokeep1;
          array    all1 {*} %name2(&tokeep1,4) %name2(&total1,4);
          array allperc1 {*} %name2(&tokeep1,5) %name2(&total3,5);
          array    gen1 {*} %name2(&tokeep1,1) %name2(&total1,1);
          array genperc1 {*} %name2(&tokeep1,2) %name2(&total1,2);
             do i = 1 to dim(items1);
           if &total1.1 not in(0, .) then genperc1{i} =  ROUND((gen1{i} / &total1.1) *
     100,.01);
           if &total1.4 not in(0, .) then allperc1{i} =  ROUND((all1{i} / &total1.4) *
     100,.01);
                    end;
```

Second look at denominators



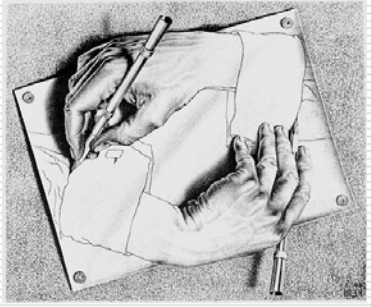
```
*DOG;  do i = 1 to dim(items1);
        if &total1.1 not in(0, .) then genperc1{i} =  ROUND((gen1{i} /
    &total1.1) * 100,.01);
        if &total1.4 not in(0, .) then allperc1{i} =  ROUND((all1{i} / &total1.4)
    * 100,.01);
                end;
*CAT;   do i = 1 to dim(items2);
        if &total2.1 not in(0, .) then genperc2{i} =  ROUND((gen2{i} /
    &total2.1) * 100,.01);
        if &total2.4 not in(0, .) then allperc2{i} =  ROUND((all{i} / &total2.4) *
    100,.01);
                end;
*PIG;   do i = 1 to dim(items6);
        if &total3.1 not in(0, .) then genperc3{i} =  ROUND((gen3{i} /
    &total6.1) * 100,.01);
        if &total3.4 not in(0, .) then allperc3{i} =  ROUND((all3{i} / &total3.4)
    * 100,.01);
                end;
```

Macro and array summary



```
%macro name2(macrovar,suf);
%let i=0;
%do %while(%scan(&macrovar,&i+1,%str( ))
    ne %str( ));
%let i = %eval(&i+1);
%let var=%scan(&macrovar,&i,%str( ));
&var.&suf
%end;
%mend name2;
```

ANY QUESTIONS ??????????????????????????



David B. Cohen, Systems Analyst

School Finance Unit Website:

http://ritter.tea.state.tx.us/school.finance/

School Finance Unit Email:

david.cohen@tea.state.tx.us

Phone Number:

(512) 475-2578

Fax Number:

(512) 936-2313