

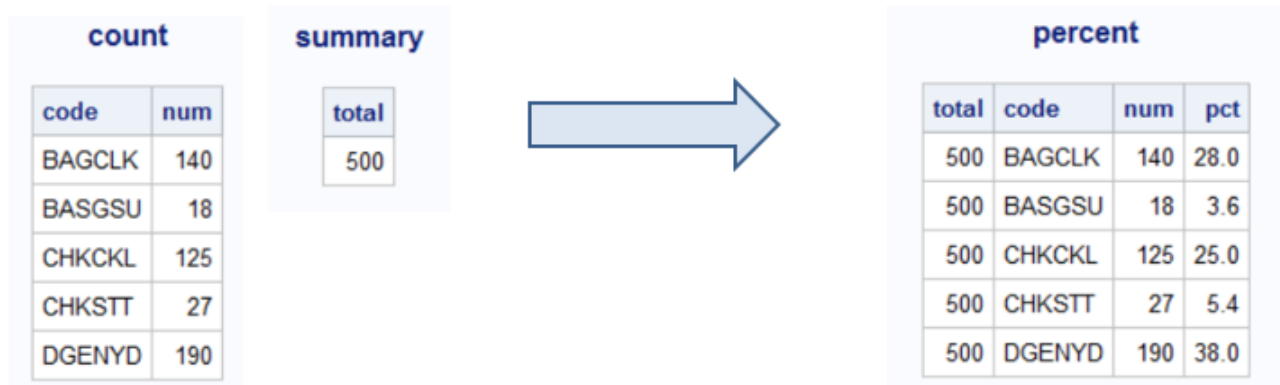
SAS training exercise

1. sas overview

assign a Libref `LIBNAME libref 'sas-data-library'<options>`

2. Understanding DATA step

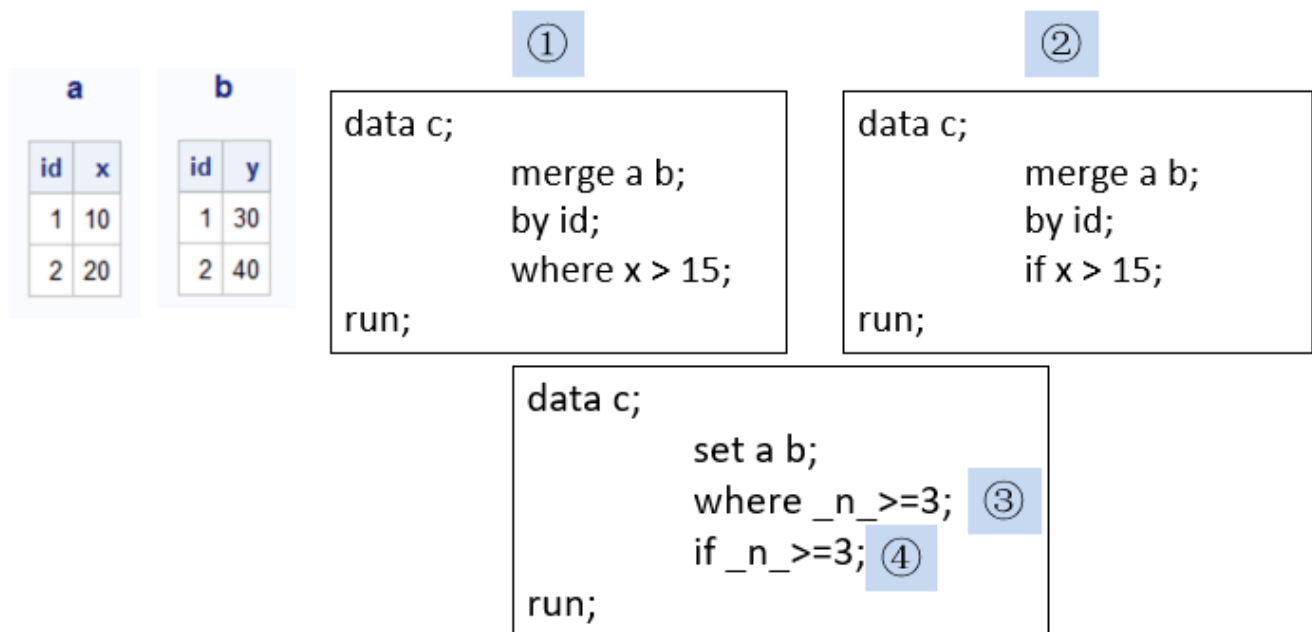
quiz1 use the two data sets('count' and 'summary') as below to output 'percent' data set on the right side.



A:

```
data percent;
  if _n_ =1 then summary;
  set count;
  pct = num/total*100;
run;
```

quiz2 Please make a judgement which of following codes is valid.



A:

1. is invalid (where clause read data before data step into pdv)
2. is valid
3. is valid
4. is invalid

if语句是面向PDV（logical program data vector）的，对当前PDV中的数据进行判断，满足条件时将其写入到外部数据集

where语句也是面向PDV的，它使用于从外部数据源读数据到PDV之前进行判断，当满足条件时才被写入到PDV。显然一个在写入PDV之前，一个在写入PDV之后，两者是有差异的。

quiz3 describe the pdv step by step.

```
data total_points (drop=TeamName);
    put _error_ ;
    input TeamName $ Name $ Event1 Event2 Event3;
    TeamTotal + (Event1 + Event2 + Event3);
    datalines;
Knights Sue      6  8  8
Cardinals Jane   9  7  8
Knights John     7  7  7
;
run;
```

3. Combining data set

quiz1. merge two data set and divide the value of variable 'weight' by 2

VITALS Dataset

SUBJID	VISIT	HEART
1	1	60
1	2	58
2	1	74
2	2	72
2	3	69
3	1	71

DEMOG Dataset

SUBJID	AGE	WEIGHT
1	42	185
2	55	170
3	30	160

```
``sas data dt; merge vital demog; by subjid; run; data final; set dt; weight=weight/2; run; ``
```

quiz2. use the ruler to combine two data set.

health				fitness	
id	name	team	weight	id	weight
1114	sally	blue	125	1114	119
1441	sue	green	145	1994	174
1750	joey	red	189	2304	170
1994	mark	yellow	165		
2304	joe	red	170		

rules : "change" is change from weight in "health" to weight in "fitness"	status
change > 0	G
change < 0	L
change = 0	E
change = .	NO WEIGHT

```

proc sort data = health;
  by id;
run;
proc sort data=fitness;
  by id;
run;
data final;
  merge health(rename=(weight=orig) in=a) fitness(in=b);
  by id;
  if a and b then do;
    if weight-orig<0 then status="L";
    else if weight-orig>0 then status="G";
    else status = "E";
  end;
  else status="NO WEIGHT";
run;

```

quiz3 Combine dataset a b and c d to generate dataset "result".

a	
id	value
1	23
5	63
7	86

b	
id	value
2	99
3	48
5	12
7	77

c	
id	value
1	42
2	86
5	41
7	17
9	76

d	
id	value
1	74
2	48
5	92
9	43



result				
id	value1	value2	value3	value4
1	23	.	42	74
2	.	99	86	48
9	.	.	76	43

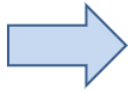
```

data result;
  merge a(in=a rename=(value=value1))
        b(in=b rename=(value=value2))
        c(in=c rename=(value=value3))
        d(in=d rename=(value=value4));
  by id;
/*result 的结果有什么规律? */
  if ^(ina and inb) and (inc and ind);
  /*if ina and inb and inc and ind*/
run;

```

4. Do-loop and Array

quiz1 transpose data "summary" to compute two variables "diff13" and "diff23"



disease	analyte	group	n	mean
DL	HDL	1	509	55.408
DL	HDL	2	303	53.191
DL	HDL	3	511	56.447
DL	LDL	1	506	101.631
DL	LDL	2	300	88.226
DL	LDL	3	506	96.541
DM	A1c	1	287	6.822
DM	A1c	2	302	7.003
DM	A1c	3	335	6.780

n1	n2	n3	mean1	mean2	mean3	disease	analyte	diff13	diff23
509	303	511	55.408	53.191	56.447	DL	HDL	1.039	3.256
506	300	506	101.631	88.226	96.541	DL	LDL	-5.090	8.315
287	302	335	6.822	7.003	6.780	DM	A1c	-0.042	-0.223

```
proc transpose data=summary out=trans1 prefix=group;
  by disease analyte;
  var n mean;
  id group;
run;

data trans2;
  merge trans1(where=(_name_="n") rename=(group1=n1 group2=n2 group3=n3))
        trans1(where=(_name_="mean") rename=(group1=mean1 group2=mean2
group3=mean3));
  by disease analyte;
  diff13=mean3-mean1;
  diff23=mean3-mean2;
run;

/* array 1 */
data array_dif1(drop= group n mean);
  array num n1-n3;
  array men mean1-mean3;
  do until (last.analyte);
    set summary;
    by disease analyte;
    num{group}=n;
    men{group}=mean;
  end;
  diff13=mean3-mean1;
  diff23=mean3-mean2;
run;

proc print data=array_dif1 noobs;title "array_dif1";run;

/* array 2 */
data array_dif2(drop= group n mean);
  array men{3,2} n1 mean1
                  n2 mean2
                  n3 mean3;
```

```

do until (last.analyte);
  set summary;
  by disease analyte;
  men{group,1}=n;
  men{group,2}=mean;
end;
diff13=mean3-mean1;
diff23=mean3-mean2;
run;

```

quiz2 Find in the missing data and replace it with the variable 'makeup'

score

subjid	time1	time2	time3	time4	time5	time6	makeup
001	0.5	0.3	0.0	0.0	0.6	0.7	0.5
002	0.1	0.5	.	.	1.1	0.0	0.4
003	0.0	0.0	1.1	.	0.0	0.9	0.3
004	0.0	0.6	0.4	.	0.0	0.0	0.7
005	0.7	0.0	0.5	.	0.5	0.5	0.2
006	0.0	1.4	0.7	0.7	.	1.0	0.7

TIME1	TIME2	TIME3	TIME4	TIME5	MAKEUP
A	B	.	D	E	C

```


/*quiz 2 */
data score;
  input subjid $ time1 time2 time3 time4 time5 time6 makeup;
  cards;
001 0.5 0.3 0.0 0.0 0.6 0.7 0.5
002 0.1 0.5 . . 1.1 0.0 0.4
003 0.0 0.0 1.1 . 0.0 0.9 0.3
004 0.0 0.6 0.4 . 0.0 0.0 0.7
005 0.7 0.0 0.5 . 0.5 0.5 0.2
006 0.0 1.4 0.7 0.7 . 1.0 0.7
;
run;
proc print data=score noobs;title "score";run;
/*如果app[i] 是缺失值的话，那么就用makeup对应的值去填充app[i]*/
data replace;
  set score;
  array apps [5] time1- time5;
  do i=1 to dim(apps);
    if apps[i] =. then apps[i]=makeup ;
    /*app[i]=ifn(app[i]=.,makeup,app[i]);*/
  end;
  drop i ;
run;

```

quiz3 Fill the missing data in data set "Score" with LOCF(Last Observation carried forward)method.

score						
subjid	time1	time2	time3	time4	time5	time6
001	0.5	0.3	0.0	0.0	0.6	0.7
002	0.1	0.5	.	.	1.1	0.0
003	0.0	0.0	1.1	.	0.0	0.9
004	0.0	0.6	0.4	.	0.0	0.0
005	0.7	0.0	0.5	.	0.5	0.5
006	0.0	1.4	0.7	0.7	.	1.0

TIME1	TIME2	TIME3	TIME4	TIME5
A	B	.	.	E



TIME1	TIME2	TIME3	TIME4	TIME5
A	B	B	B	E

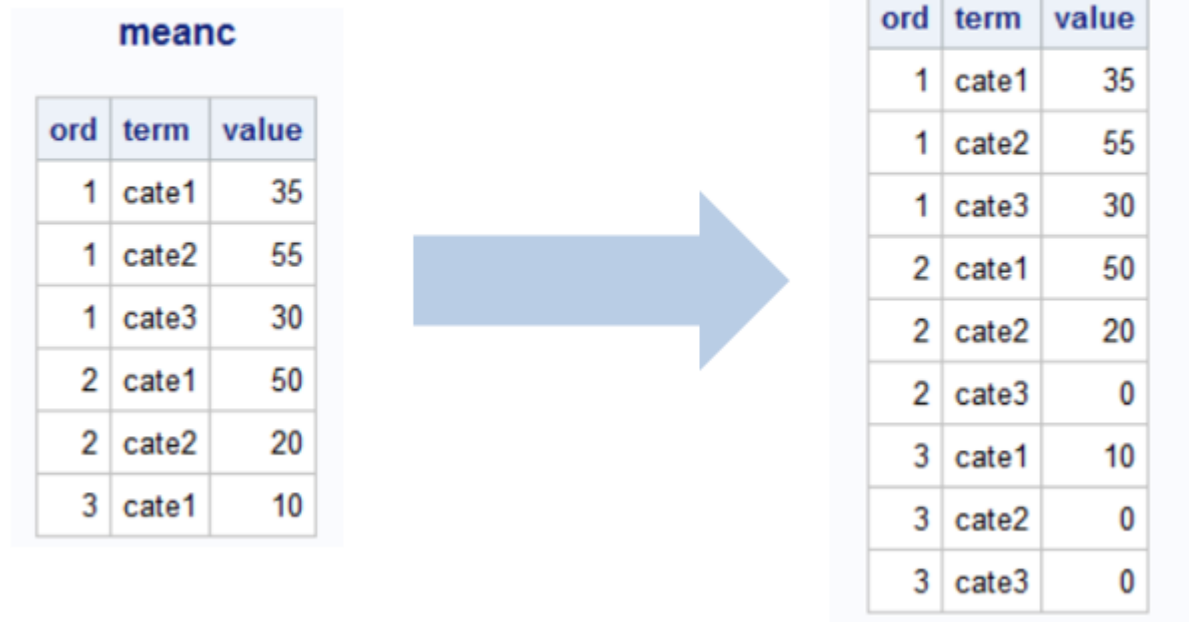
```

/*quiz 3 */
data score;
    input subjid $ time1 time2 time3 time4 time5 time6;
    cards;
001 0.5 0.3 0.0 0.0 0.6 0.7
002 0.1 0.5 . . 1.1 0.0
003 0.0 0.0 1.1 . 0.0 0.9
004 0.0 0.6 0.4 . 0.0 0.0
005 0.7 0.0 0.5 . 0.5 0.5
006 0.0 1.4 0.7 0.7 . 1.0
;
run;

data locf(drop=i);
    set score;
    array time(*) time1-time6;
    do i=1 to dim(time);
        /*实际上不是真正的 LOCF 而且这个程序有个bug 即第一列的数据不能为缺失值*/
        if time(i)=. then time(i)=time(i-1);
    end;
run;

```

quiz4 Add temporary missing categories for data set "meanc" to generate data set "result" as below.



```

data meanc;
    input ord term $ value;
    cards;
1 cate1 35
1 cate2 55
1 cate3 30
2 cate1 50
2 cate2 20
3 cate1 10
;
run;
proc print data=meanc noobs;title "meanc";run;

data dummy;
    array cate{3} $ _temporary_ ('cate1' 'cate2' 'cate3');
    do ord=1 to 3;
        do i=1 to dim(cate);
            term=cate{i};value=0;output;
        end;
    end;
    drop i;
run;

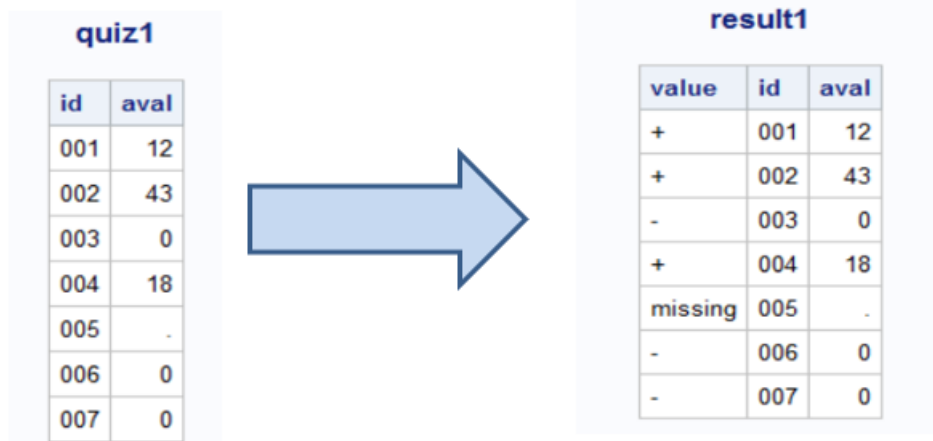
data result;
    update dummy(in=ina) meanc(in=inb);
    by ord term;
run;

```

5. sas Function

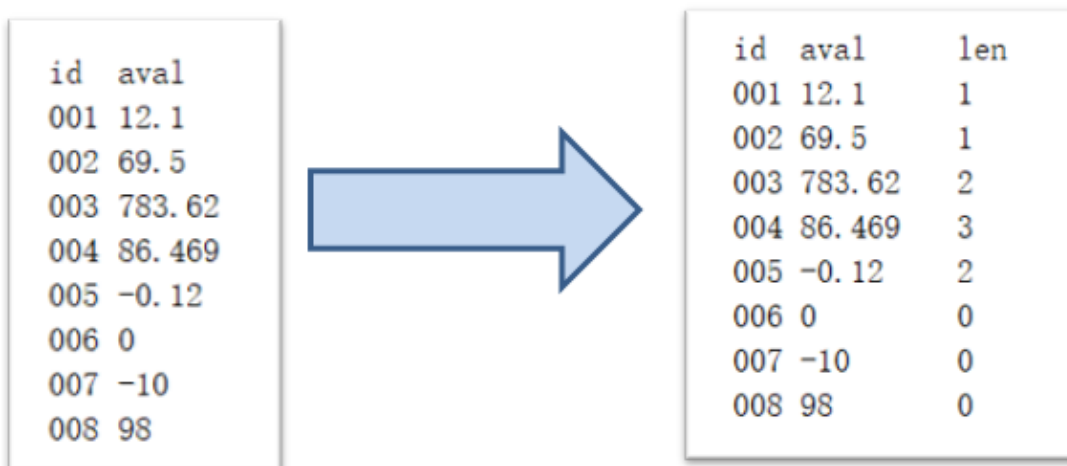
quiz1 Use the function to calculate the variable "value" according to SAS data set "quiz1" and algorithm as below.

Aval>0	Value="+"
Aval=0	Value="-"
Aval=.	Value="missing"



```
data result1;
  length value $10.;
  set quiz1;
  value=ifc(Aval,"+","-","missing");
  /*    if aval>0 then value="+";*/
  /*    else if aval=0 then value="-";*/
  /*    else if aval=. then value="missing";*/
```

quiz2 Use to function to calculate decimal place of variable "aval" .




```

data quiz2;
    input id$ aval;
    aval=aval;
    cards;
001 12.1
002 69.5
003 783.62
004 86.469
005 -0.12
006 0
007 -10
008 98
;
run;

data result2;
    set quiz2;
    len=ifn(aval ne int(aval),length(left(put(abs(aval),best.)))-
length(left(put(int(abs(aval)),best.)))-1,0);
    /*aval ne int(aval) 指的是aval 是否有小数部分
    left(put(abs(aval),best.)) 取aval得绝对值 并将其置为best格式 SAS 自动选择最合
    适的数字格式 (w 缺省值位 12) 右对齐
    left() 除去左边的空格 length()计算长度*/
run;

```

quiz2 use function to calculate the demical place of variable "aval".



id	aval
001	12.1
002	69.5
003	783.62
004	86.469
005	-0.12
006	0
007	-10
008	98

id	aval	len
001	12.1	1
002	69.5	1
003	783.62	2
004	86.469	3
005	-0.12	2
006	0	0
007	-10	0
008	98	0

```

data quiz2;
    input id$ aval;
    aval=aval;
    cards;
001 12.1
002 69.5

```

```

003 783.62
004 86.469
005 -0.12
006 0
007 -10
008 98
;
run;

data result2;
    set quiz2;
    len=ifn(aval ne int(aval),length(left(put(abs(aval),best.)))-
length(left(put(int(abs(aval)),best.)))-1,0);
/*aval ne int(aval) 指的是aval 是否有小数部分
left(put(abs(aval),best.))
取aval得绝对值 并将其置为best格式 SAS 自动选择最合适的数字格式（w 缺省值位 12） 右对齐
left() 除去左边的空格 length()计算长度
*/
run;

```

quiz3 output the following cotents in the log according to sas functions learned.

我出生在xxxx 年的第x个季度,也是一年中的第x 个月,也是一年中的第x 周的一个周x 。这一天是这个月的第x 天。今天是： xxxx-xx-xx ,我来到这世界已经xx 年了。准确的来说我已经活了xxxx 天。如果我能活100岁,那么我还能活xxxxx 天。我比SAS日期的生日（SAS日期开始计算的时间"1960-01-01T00:00"）晚了xxxxx 天

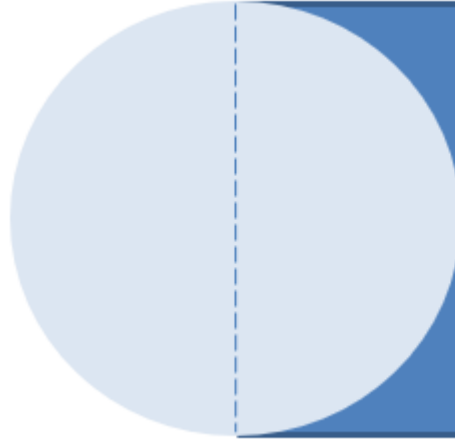
```

data __null__;
    format today yymmdd10.;
    birthday='07jul1991'd;
    byear=year(birthday);
    bqtr=qtr(birthday);
    bmonth=month(birthday);
    bweek=week(birthday);
    bweekd=weekday(birthday)-1;
    bday=day(birthday);
    today=today();
    ageYear=intck('year',birthday,today);
    ageDay=intck('day',birthday,today);
    dayto100=365*100-ageyear;
    daylate=birthday;
put "我出生在" byear "的第" bqtr "个季度,也是一年中的第" bmonth "个月,也是一年中的第"
bweek "周的一个周" bweekd "。这一天是这一个月第" bday "天。今天是: " today ",我来到
这个世界已经" ageYear "年了。准确的来说我已经活了" ageDay "天。如果我能活到100岁,那么
我还能活" dayto100 "天。我比sas日期的生日（sas日期开始计算的时间"1960-01-01T00:00:
00"）晚了" dayLate "天";
/*
我出生在1991 的第3 个季度,也是一年中的第7 个月,也是一年中的第27 周的一个周1。
这一天是这一个月第7 天。今天是: 2019-09-10,我来到这个世界已经28 年了。
准确的来说我已经活了10292 天。如果我能活到100岁,那么我还能活36472天。
我比sas日期的生日（sas日期开始计算的时间"1960-01-01T00:00: 00"）晚了11510 天

```

```
*/
run;
```

quiz4 The area of a given rectangle is 18. Please calculate the area of the shaded area.



```
data _null_;
    radius=sqrt(18/2);
    halfCircleArea=1/2*constant('pi')*radius**2;
    shadedArea=18-halfCircleArea;
    put "the radius of the circle is " radius "the area of half circle
is:"halfCircleArea; put "the area of the shade is "shadedArea;
run;

/*
the radius of the circle is 3 the area of half circle is:14.137166941
the area of the shade is 3.8628330588
*/
```

SAS Fucntion Summary

ABS(x)	求x的绝对值。
MAX(x1,x2,...,xn)	求所有自变量中的最大值。
MIN(x1,x2,...,xn)	求所有自变量中的最小值。
MOD(x,y)	求x除以y的余数。
SQRT(x)	求x的平方根。
ROUND(x,eps)	求x按照eps指定的精度四舍五入后的结果，比如
ROUND(5654.5654,0.01)	=5654.57
ROUND(5654.5654,10)	= 5650。
CEIL(x)	求大于等于x的最小整数。当x为整数时就是x本身，否则为x右边最近的整数。
FLOOR(x)	求小于等于x的最大整数。当x为整数时就是x本身，否则为x左边最近的

整数。	
INT(x)	求x扔掉小数部分后的结果。
FUZZ(x)	当x与其四舍五入整数值相差小于1E-12时取四舍五入。
LOG(x)	求x的自然对数。
LOG10(x)	求x的以10为底的对数。
EXP(x)	指数函数。
SIN(x)	求x的正弦。
COS(x)	求x的余弦。
TAN(x)	求x的正切。
ARSIN(y)	计算函数 $y=\sin(x)$ 在区间的反函数，y取[-1,1]间值。
ARCOS(y)	计算函数 $y=\cos(x)$ 在的反函数，y取[-1,1]间值。
ATAN(y)	计算函数 $y=\tan(x)$ 在的反函数，y取间值。
SINH(x)	双曲正弦
COSH(x)	双曲余弦
TANH(x)	双曲正切
ERF(x)	误差函数
GAMMA(x)	完全函数

此外还有

符号函数SIGN
 函数一阶导数函数DIGAMMA
 二阶导数函数TRIGAMMA
 误差函数余函数ERFC
 函数自然对数LGAMMA
 ORDINAL函数
 AIRY 函数
 DAIRY函数
 Bessel函数JBESSEL
 修正的Bessel函数IBESSEL，等。

/*数组函数*/

/*数组函数计算数组的维数、上下界，有利于写出可移植的程序。数组函数包括：*/

DIM(x) 求数组x第一维的元素的个数（注意当下界为1时元素个数与上界相同，否则元素个数不一定与上界相同）。

DIM k(x) 求数组x第k维的元素的个数。

LBOUND(x) 求数组x第一维的下界。

HBOUND(x) 求数组x第一维的上界。

LBOUND k(x) 求数组x第 k维的下界。

HBOUND k(x) 求数组x第 k维的上界。

/*字符函数*/

PUT(X, ??best.) 数值型转换成字符型(?? 作为一种模糊识别符号，在任何format中都可以使用，作用可以避免报错)。

INPUT(X) 字符型转换成数值型。

TRIM(s) 返回去掉字符串s的尾随空格的结果。

STRIP(s) 去掉首尾空格。

UPCASE(s) 把字符串s中所有小写字母转换为大写字母后的结果。

LOWCASE(s) 把字符串s中所有大写字母转换为小写字母后的结果。

PROPCASE(s) 首字母大写。

INDEX(s,s1) 查找s1在s中出现的位置。找不到时返回0。

FIND(character-value, find-string <,'modifiers'> <,start>)

/*检索”字符串“中的”子字符串“。可附加选项：指定检索开始位置；检索方向；忽略大小写或末尾空格。*/

SUBSTR(s,p,n) 从字符串s中的第p个字符开始抽取n个字符长的子串。

TRANWRD(s,s1,s2) 从字符串s中把所有字符串s1替换成字符串s2后的结果。

TRANSLATE(source, to-1, from-1<,...to-n, from-n>) 替换字符表达式中的特定字符。

COMPRESS(source<,chars>)删除指定字符（若不指定要删除的字符，则删除string中的全部空格）。例如compress(subject_addtl_id,".", "kd"), 只保留'.'和数字（"kd"表示“keep digits only”）。

SCAN(X, n<,>delimiters>) 返回字符表达式X中的第n个词。

COALESCE(S) 挑选首个遇到的不缺失的值(适用于数值型变量)。

COALESCEC(S) 挑选首个遇到的不缺失的值(适用于字符型变量)。

/*cat系列函数（CAT/CATS/CATT/CATX）*/

CAT函数 “cat”代表“concatenate”，等价于【a||b】仅仅是单纯地串联字符串，不进行任何其他操作。

CATS函数 “s”代表“strip”，等价于【strip(a)||strip(b)】或【TRIM(LEFT(a))||TRIM(LEFT(b))】将各个字符串去掉首尾空格后再进行串联。

CATT “t”代表“trim”，等价于【TRIM(a)||TRIM(b)】将各个字符串去掉尾部空格后再进行串联。

CATX函数 等价于【strip(a)||”-”||strip(b)】或【TRIM(LEFT(a))||”-”||TRIM(LEFT(b))】

将各个字符串去掉首尾空格后再用指定的连接符进行串联。

IFN 根据表达式是否为真、假或缺失返回一个数值(适用于数值型)。

IFC 根据表达式是否为真、假或缺失返回一个字符值(适用于字符型)。

RANK(s) 字符s的ASCII码值。

BYTE(n) 第n个ASCII码值的对应字符。

REPEAT(s,n) 字符表达式s重复n次。

/*时间日期函数*/

MDY(m,d,yr) 生成yr年m月d日的SAS日期值

YEAR(date) 由SAS日期值date得到年

MONTH(date) 由SAS日期值date得到月

DAY(date) 由SAS日期值date得到日

WEEKDAY(date) 由SAS日期值date得到星期几

QTR(date) 由SAS日期值date得到季度值

HMS(h,m,s) 由小时h、分钟m、秒s生成SAS时间值

DHMS(d,h,m,s) 由SAS日期值d、小时h、分钟m、秒s生成SAS日期时间值

DATEPART(dt) 求SAS日期时间值dt的日期部分

INTNX(interval,from,n)

计算从from开始经过n个in间隔后的SAS日期。

其中interval 可以取'YEAR'、'QTR'、'MONTH'、'WEEK'、'DAY'等。比如，INTNX('MONTH', '16Dec1997'd, 3)结果为1998年3月1日。注意它总是返回一个周期的开始值。

INTCK(interval,from,to)

计算从日期from到日期to中间经过的interval间隔的个数，其中interval取'MONTH'等。

比如，INTCK('YEAR', '31Dec1996'd, '1Jan1998'd)计算1996年12月31日到1998年1月1日经过的年间隔的个数，结果得2，尽管这两个日期之间实际只隔1年。

6. Macro

quiz1 count sashelp.air observation number and save it into macro variable 'nobs';

```
proc sql noprint;
    select count(*) into : nobs from sashelp.air;
quit;

data _null_;
    result=resolve('the number of observations are &nobs');
    put result;
run;
/*The number of observations are          19*/
```

quiz2 use call symput or proc sql to create macro variables:

grpNum: Number of treatment groups

g1Name - g&grpNum.Name : Treatment Name for each group

g1Num - g&grpNum.Num : Subject Number in each group;

```
data trt;
do id=1 to 100;
    trtn=(uniform(int(time()))>0.5)+1;
    trt="drug"||put(trtn,1.);
    output;
end;
run;

%macro Q2(ds);
proc sort data=&ds.;
by trtn trt;
run;
data test;
    retain grptot;
    set &ds.;
    by trtn trt;
    if first.trtn then grptot=1;
    else grptot+1;
    if last.trtn then output;
run;

data _null_;
    set test end=last;
    call symput('g' || strip(put(_n_,best.)) || 'Name',trt);
    call symputx('g' || strip(put(_n_,best.)) || 'Num',put(grptot,best.));
    if last then call symputx('grpNum',put(_n_,best.));
run;
%do grpNum=1 %to &grpNum.;
%put The value of Macro Variable g&grpNum.Name is &&g&grpNum.Name;
```



```

    %put The value of Macro Variable g&grpNum.Num is &&g&grpNum.Num;
  %end;
%mend Q2;

%Q2(trt)

```

quiz3 define a macro function implement the same function as 2. with input dataset as parameter.
To ensure these macro variables are valid outside of the function;

```
%Q2(trt)
```

quiz4. We have below codes, Please complete macro test 2 to output results below in log file.

```

%macro test1;
    %put <-----it is test1!;
%mend test1;
%macro test2 (mac);
    %put<-----test2 start!;
    * <-----add codes here!!!
    %&mac.;
    %put <-----test2 end!;
%mend test2;
%test2(test1);
/*
Outputs in log:
<-----test2 start!
<-----it is test1!
<-----test2 end!
*/

```

quiz5

```

%let message=;
%put &message;
*log output: Bob's father said "it's P&G company's product.";

```

```

%let message = %bquote(Bob's father said "it's P&G company's product.");

/* %let message = %nrstr(Bob's father said "it's P&G company's product."); */

%put &message;

```

macro function

```

call symputx('a',5);
/*宏函数与字符串*/
*包含特殊字符;
%str(mtest age,age-sex/canprint;)
*求值;
%let sum=1+1; %put &sum; *1+1;
%let total=%eval(1+1); %put &total; *2;

/*内置宏函数*/
%STR and %NRSTR
%BQUOTE and %NRBQUOTE
%SUPERQ
%upcase, %index, %scan
%eval, %sysfunc
/*eval函数只能做整型的书 相加，当出现小数的时候 就会出错，
这个时候需要使用%syseval()函数*/

/*Q开头的宏函数*/
%SCAN and %QSCAN
%SUBSTR and %QSUBSTR
%UPCASE and %QUPCASE
%SYSFUNC and %QSYSFUNC

/*
当字符串中出现& 和% 的时候用函数%NRSTR、%NRBQUOTE、%SUPERQ、%NRQUOTE
当字符串中出现不成对的'和"时，用函数
*/
data-variable = symget('macro-variable')

data _null_;
    call symput('symput',5);
    call symputx('symputx',5);
run;
%put |&symput|;
%put |&symputx|;
/*
|      5 |
|5|
*/

```