



BC Cancer Agency

CARE + RESEARCH

An agency of the Provincial Health Services Authority

Cancer Surveillance & Outcomes

Tips for SAS Programming

Yaling Yin

Cancer Surveillance & Outcomes (CSO)

Population Oncology

BC Cancer Agency

Outlines

- IFN and IFC functions
- LAG and DIF functions
- Obtain a data set with current and next rows
- Obtain a data set with current, previous and next rows by groups
- SAS 9.4 used



IFN / IFC Function

- IFN/IFC
 - (condition,
 - value1 if condition is TRUE,
 - value2 if condition is FALSE,
 - <value3 if condition is MISSING>)
- If argument 4 is not specified then value2 is returned for missing condition



IFN / IFC Function

- A short version of IF-THEN-ELSE
- IFN - returns numeric values
- IFC - returns character values



IFN / IFC Function

```
data monthly_records;
  input ID $ 1-3 gender $ 5 visit_in_April 7-8
        married 10;
  CARDS;
001 M 3 1
002 F 2 0
003 F 1
004 M 1 0
005 M
006 M 10
;
run;
```



IFN / IFC Function

```
data monthly_records;
  length marriage_status $8;
  set monthly_records;
  if married=1 then marriage_status='married';
  else marriage_status='single';
run;
```

```
data monthly_records;
  length marriage_status $8;
  set monthly_records;
  marriage_status=ifc(married=1, 'married', 'single');
run;
```



IFN / IFC Function

	ID	gender	visit_in_April	married	marriage_status
1	001	M	3	1	married
2	002	F	2	0	single
3	003	F	.	1	married
4	004	M	1	0	single
5	005	M	.	.	single
6	006	M	10	.	single

- married=. is evaluated as 'FALSE' when the optional argument is missing
- If the length of marriage_status is not specified, IFC will assign it a length of 200 by default.



IFN / IFC Function

```
data monthly_records;  
  set monthly_records;  
  marriage_status=ifc(married=1, 'married', 'single', 'unknown');  
run;
```

	ID	gender	visit_in_April	married	marriage_status
1	001	M	3	1	married
2	002	F	2	0	single
3	003	F	.	1	married
4	004	M	1	0	single
5	005	M	.	.	single
6	006	M	10	.	single

- If married is a missing value then 'single' is returned, even when the optional argument is specified.



IFN / IFC Function

```
data monthly_records;
  length marriage_status $8;
  set monthly_records;
  marriage_status=ifc(married, 'married', 'single', 'unknown');
run;
```

```
data monthly_records;
  length marriage_status $8;
  set monthly_records;
  if married=1 then marriage_status='married';
  else if married=0 then marriage_status='single';
  else marriage_status='unknown';
run;
```



IFN / IFC Function

	ID	gender	visit_in_April	married	marriage_status
1	001	M	3	1	married
2	002	F	2	0	single
3	003	F	.	1	married
4	004	M	1	0	single
5	005	M	.	.	unknown
6	006	M	10	.	unknown

- If married is a missing value then 'unknown' is returned
- By default, SAS evaluate numeric value as:
 - FALSE – 0
 - TRUE – 1 or values other than 0 or missing
 - MISSING – .



IFN / IFC Function

```
data monthly_records;  
  length marriage_status $8;  
  set monthly_records;  
  marriage_status=ifc(married=1, 'married',  
    ifc(married=., 'unknown', 'single'));  
run;
```

```
data monthly_records;  
  length marriage_status $8;  
  set monthly_records;  
  marriage_status=ifc(married, 'married', 'single', 'unknown');  
run;
```

```
data monthly_records;  
  length marriage_status $8;  
  set monthly_records;  
  if married=1 then marriage_status='married';  
  else if married=0 then marriage_status='single';  
  else marriage_status='unknown';  
run;
```



IFN / IFC Function

- Difference between IF-THEN-ELSE and IFN/IFC
 - Default lengths of returned variables
 - When returned value is based on a formula, IF-THEN-ELSE evaluates the condition first then does the calculation, whereas IFN/IFC does the opposite.



IFN / IFC Function

```
data dat;  
    input x 1 y 3-5;  
    CARDS;  
5 200  
1 100  
0 150  
2 300  
;  
run;
```

```
data new_dat;  
    set dat;  
    if x^=0 then z=y/x;  
    else z=y;  
run;
```



IFN / IFC Function

	x	y	z
1	5	20	4
2	1	10	10
3	0	15	15
4	2	30	15

NOTE: There were 4 observations read from the data set WORK.DAT.
NOTE: The data set WORK.NEW_DAT has 4 observations and 3 variables.
NOTE: DATA statement used (Total process time):
real time 0.01 seconds
cpu time 0.01 seconds



IFN / IFC Function

```
data new_dat;  
  set dat;  
  new=ifn(x^=0, y/x, y);  
run;
```

	x	y	new
1	5	20	4
2	1	10	10
3	0	15	15
4	2	30	15

NOTE: Division by zero detected at line 806 column 19.

x=0 y=15 new=15 _ERROR_=1 _N_=3

NOTE: Mathematical operations could not be performed at the following places. The results of the operations have been set to missing values.

Each place is given by: (Number of times) at (Line):(Column).
1 at 806:19

NOTE: There were 4 observations read from the data set WORK.DAT.

NOTE: The data set WORK.NEW_DAT has 4 observations and 3 variables.

NOTE: DATA statement used (Total process time):

real time 0.01 seconds

cpu time 0.01 seconds



LAG and DIF Functions

Example: Sashelp.fish

	Species	Weight	Length1	Length2	Length3	Height	Width
139	Roach	390	29.5	31.7	35	9.485	5.355
140	Smelt	6.7	9.3	9.8	10.8	1.7388	1.0476
141	Smelt	7.5	10	10.5	11.6	1.972	1.16
142	Smelt	7	10.1	10.6	11.6	1.7284	1.1484
143	Smelt	9.7	10.4	11	12	2.196	1.38
144	Smelt	9.8	10.7	11.2	12.4	2.0832	1.2772
145	Smelt	8.7	10.8	11.3	12.6	1.9782	1.2852
146	Smelt	10	11.3	11.8	13.1	2.2139	1.2838
147	Smelt	9.9	11.3	11.8	13.1	2.2139	1.1659
148	Smelt	9.8	11.4	12	13.2	2.2044	1.1484
149	Smelt	12.2	11.5	12.2	13.4	2.0904	1.3936
150	Smelt	13.4	11.7	12.4	13.5	2.43	1.269
151	Smelt	12.2	12.1	13	13.8	2.277	1.2558
152	Smelt	19.7	13.2	14.3	15.2	2.8728	2.0672
153	Smelt	19.9	13.8	15	16.2	2.9322	1.8792
154	Whitefish	270	23.6	26	28.7	8.3804	4.2476
155	Whitefish	270	24.1	26.5	29.3	8.1454	4.2485
156	Whitefish	306	25.6	28	30.8	8.778	4.6816
157	Whitefish	540	28.5	31	34	10.744	6.562
158	Whitefish	800	33.7	36.4	39.6	11.7612	6.5736
159	Whitefish	1000	37.3	40	43.5	12.354	6.525



LAG and DIF Functions

```
❑ data fish_weight;  
    set sashelp.fish;  
    keep species weight;  
run;  
-----  
❑ proc sort data=fish_weight;  
    by species weight;  
run;  
-----
```

	Species	Weight
139	Roach	390
140	Smelt	6.7
141	Smelt	7
142	Smelt	7.5
143	Smelt	8.7
144	Smelt	9.7
145	Smelt	9.8
146	Smelt	9.8
147	Smelt	9.9
148	Smelt	10
149	Smelt	12.2
150	Smelt	12.2
151	Smelt	13.4
152	Smelt	19.7
153	Smelt	19.9
154	Whitefish	270
155	Whitefish	270
156	Whitefish	306
157	Whitefish	540
158	Whitefish	800
159	Whitefish	1000



LAG and DIF Functions

- LAG – returns the values of previous rows of a variable(column)
- DIF – returns the difference between current rows and previous rows

```
data prev_fish;  
  set fish_weight;  
  Prev_weight = lag(weight);  
  Diff_weight = dif(weight);  
run;
```

	Species	Weight	Prev_weight	Diff_weight
139	Roach	390	290	100
140	Smelt	6.7	390	-383.3
141	Smelt	7	6.7	0.3
142	Smelt	7.5	7	0.5
143	Smelt	8.7	7.5	1.2
144	Smelt	9.7	8.7	1
145	Smelt	9.8	9.7	0.1
146	Smelt	9.8	9.8	0
147	Smelt	9.9	9.8	0.1
148	Smelt	10	9.9	0.1
149	Smelt	12.2	10	2.2
150	Smelt	12.2	12.2	0
151	Smelt	13.4	12.2	1.2
152	Smelt	19.7	13.4	6.3
153	Smelt	19.9	19.7	0.2
154	Whitefish	270	19.9	250.1
155	Whitefish	270	270	0
156	Whitefish	306	270	36
157	Whitefish	540	306	234
158	Whitefish	800	540	260
159	Whitefish	1000	800	200



Obtain the Values of Previous Rows by groups

```
data prev_fish;  
  set fish_weight;  
  by species weight;  
  Prev_weight = ifn( first.species, (.), lag(weight) );  
  Diff_weight = ifn( first.species, (.), dif(weight) );  
run;
```

- Obtain a data set with current rows, previous rows and the difference of the two rows for each species
- The IFN function sets the first Prev_weight and DIFF_weight of each species to be 0.
- The BY statement is used to generate automatic variable first.species.



Obtain the Values of Previous Rows by groups

	Species	Weight	Prev_weight	Diff_weight
140	Smelt	6.7	.	.
141	Smelt	7	6.7	0.3
142	Smelt	7.5	7	0.5
143	Smelt	8.7	7.5	1.2
144	Smelt	9.7	8.7	1
145	Smelt	9.8	9.7	0.1
146	Smelt	9.8	9.8	0
147	Smelt	9.9	9.8	0.1
148	Smelt	10	9.9	0.1
149	Smelt	12.2	10	2.2
150	Smelt	12.2	12.2	0
151	Smelt	13.4	12.2	1.2
152	Smelt	19.7	13.4	6.3
153	Smelt	19.9	19.7	0.2
154	Whitefish	270	.	.
155	Whitefish	270	270	0
156	Whitefish	306	270	36
157	Whitefish	540	306	234
158	Whitefish	800	540	260
159	Whitefish	1000	800	200

```
ifn( first.species, (.), lag(weight) );
```

```
ifn( first.species, (.), dif(weight) );
```



Obtain the Values of Next Rows by Groups

```
data next_fish;
  set fish_weight;
  by species weight;
  set fish_weight (firstobs=2 keep=weight rename=(weight=Next_weight))
    fish_weight( obs = 1 drop = _all_ );
  Next_weight = ifn( last.species, (.), Next_weight );
run;
```

- The 2nd SET statement generate a data set with one column of Next_weight
- In the 2nd SET statement
 - the 1st FISH_WEIGHT contains a column shifting WEIGHT one row up
 - the 2nd FISH_WEIGHT contains only one missing value
- The BY statement is used to generate automatic variable first.species and last.species.
- The IFN function sets the last Prev_weight and DIFF_weight of each species to be 0.



Obtain the Values of Next Rows by Groups

	Species	Weight	Next_weight
140	Smelt	6.7	7
141	Smelt	7	7.5
142	Smelt	7.5	8.7
143	Smelt	8.7	9.7
144	Smelt	9.7	9.8
145	Smelt	9.8	9.8
146	Smelt	9.8	9.9
147	Smelt	9.9	10
148	Smelt	10	12.2
149	Smelt	12.2	12.2
150	Smelt	12.2	13.4
151	Smelt	13.4	19.7
152	Smelt	19.7	19.9
153	Smelt	19.9	.
154	Whitefish	270	270
155	Whitefish	270	306
156	Whitefish	306	540
157	Whitefish	540	800
158	Whitefish	800	1000
159	Whitefish	1000	.

```
data next_fish;  
  set fish_weight;  
  by species weight;  
  set
```

```
    fish_weight  
    (firstobs=2 keep=weight  
     rename=(weight=Next_weight))
```

```
    fish_weight  
    (obs = 1 drop = _all_ );
```

```
Next_weight =  
  ifn( last.species, (.),  
       Next_weight );
```

```
run;
```



Obtain a Data set of Current, Previous and Next Rows by Groups

```
data new_fish;
  set fish_weight;
  by species weight;
  set fish_weight (firstobs=2 keep=weight rename=(weight=Next_weight))
    fish_weight( obs = 1 drop = _all_ );
  Prev_weight = ifn( first.species, (.), lag(weight) );
  Diff_weight = ifn( first.species, (.), dif(weight) );
  Next_weight = ifn( last.species, (.), Next_weight );
run;
```



Obtain a Data set of Current, Previous and Next Rows by Groups

	Species	Weight	Next_weight	Prev_weight	Diff_weight
140	Smelt	6.7	7	.	.
141	Smelt	7	7.5	6.7	0.3
142	Smelt	7.5	8.7	7	0.5
143	Smelt	8.7	9.7	7.5	1.2
144	Smelt	9.7	9.8	8.7	1
145	Smelt	9.8	9.8	9.7	0.1
146	Smelt	9.8	9.9	9.8	0
147	Smelt	9.9	10	9.8	0.1
148	Smelt	10	12.2	9.9	0.1
149	Smelt	12.2	12.2	10	2.2
150	Smelt	12.2	13.4	12.2	0
151	Smelt	13.4	19.7	12.2	1.2
152	Smelt	19.7	19.9	13.4	6.3
153	Smelt	19.9	.	19.7	0.2
154	Whitefish	270	270	.	.
155	Whitefish	270	306	270	0
156	Whitefish	306	540	270	36
157	Whitefish	540	800	306	234
158	Whitefish	800	1000	540	260
159	Whitefish	1000	.	800	200



Thank you!



BC Cancer Agency

CARE + RESEARCH

An agency of the Provincial Health Services Authority

Cancer Surveillance & Outcomes