SESUG Paper 149-2020

The Efficient Alternative to LAG Function without LAG's Pitfalls

Alexander M. Feigin, Alex MedStat, Inc.

ABSTRACT

When SAS® data set is ordered by group variable, the use of LAG function(s) without combining in one observation the values from different BY-groups could be a difficult task. Instead we suggest creating separate data sets where a new _N_ is equal to _N_+1, or _N_+2, etc., and merging them together, thus in each observation combining the values from as many as needed preceding records belonging to the same BY-group only.

INTRODUCTION

LAG function can have some problems when we deal with data sets stratified by BY-group. For example, in the data set ONE_BY sorted by variable ORDO, we need to create two variables LAG1_X and LAG2_X which have values of X from one and two preceding observations respectively, inside the same BY-group ORDO.

USING LAG FUNCTION WITHOUT ANY "TRICKS" IN THE STRATIFIED DATA SET

Table 1. Data set ONE_BY stratified by the variable ORDO

Obs	x	ordo
1	75	0
2	32	0
3	18	0
4	91	0
5	36	0
6	22	0
7	79	1
8	40	1
9	12	1
10	19	1
11	78	1
12	44	1
13	97	1
14	26	2
15	71	2
16	55	2
17	53	2
18	86	2
19	14	2
20	86	2

First, let us try to use LAG function without any 'tricks' to get the previous value of X inside the data set ONE_BY stratified by ORDO.

```
data laggi1;
  set one_by;
  by ordo;
  lag1_x=lag1(x);
run;
```

We will have the following output (Table 2):

Table 2. LAG1 without any "tricks" in the stratified data set

Obs	x	ordo	lag1_x
1	75	0	.
2	32	0	75
3	18	0	32
4	91	0	18
5	36	0	91
6	**22**	0	36
7	79	1	**22**
8	40	1	79
9	12	1	40
10	19	1	12
11	78	1	19
……			

Obviously in the observation #7 the value of LAG1_X is not what we expect - it comes from the last observations (#6) in ORDO=0 group though observation #7 is the first one in ORDO=1 group (i.e. it does not have any preceding records belonging to the same ORDO group, so in the observation #7 the variable LAG1_X should have a missing value).

USING LAG FUNCTION WITH "TRICKS" IN THE STRATIFIED DATA SET

Certainly we can easily obtain the needed result by saying "if first ordo then lag1_x is equal to missing":

```
data laggi2;
  set one_by;
  by ordo;
  lag1_x=lag1(x);
  if first.ordo then lag1_x=. ;
run;
```

We have to make sure that LAG is not called in conditional code, like " if not first.ordo then lag1_x=lag1(x)" because in this case we can easily get incorrect results [1,2].

Even easier we can get the same result using IFN function instead of IF...THEN ...

```
data laggi2a;
  set one_by;
  by ordo;
  lag1_x = ifn(first.ordo, ., lag(x));
run;
```

Table 3. LAG1 with a small trick in the stratified data set

Obs	x	ordo	lag1_x
1	75	0	.
2	32	0	75
3	18	0	32
4	91	0	18
5	36	0	91
6	22	0	36
7	79	1	.
8	40	1	79
9	12	1	40
10	19	1	12
11	78	1	19
12	44	1	78
13	97	1	44
14	26	2	.
15	71	2	26
16	55	2	71
17	53	2	55
18	86	2	53
19	14	2	86
20	86	2	14
.....			

Now all observations with first.ordo=1 have missing value of LAG1_X.

USING BOTH, LAG1 AND LAG2 FUNCTION IN STRATIFIED DATA SET

What if we need to use observations from more than one-step back? We should use LAG2, LAG3, LAG4, etc. on the data set ONE_BY stratified by ORDO. Let us try with adding just LAG2:

```
data laggi3;
  set one_by;
  by ordo;
  lag1_x=lag(x);
  lag2_x=lag2(x);
  if first.ordo then
  do;
    lag1_x=. ;
    lag2_x=. ;
```

```
  end;
run;
```

Table 4. LAG1 and LAG2 with a small "trick" in the stratified data set.

Obs	x	ord	lag1_x	lag2_x
1	75	0		
2	32	0	75	
3	18	0	32	75
4	91	0	18	32
5	36	0	91	18
6	**22**	0	36	91
7	79	1	.	.
8	40	1	79	**22**
9	12	1	40	79
10	19	1	12	40
11	78	1	19	12
12	44	1	78	19
13	97	1	44	78
14	26	2		
15	71	2	26	97
16	55	2	71	26
17	53	2	55	71
18	86	2	53	55
19	14	2	86	53
20	86	2	14	86

This time observation #7 is fine, both LAG1_X and LAG2_X have missing values (similar for observations ## 14, 21, etc.). But observation #8, for example, has value for LAG2_X equal to X in the observation 6, which is the last observation in ORDO=0 group, so LAG2_X in observation 8 (which belongs to ORDO=1 group) should also have a missing value . How to bypass this problem?

CREATING ONE MORE ORDERING VARIABLE

Observation 8 in LAGGI4 data set is a second observation in ORDO=1 group. In SAS there is no way to write something like "if second.ordo then lag2_x=." as we did for the first observation in BY group. Instead we can create one more ordering variable - ORDO2, which will sort observations by ORDO2 inside each ORDO group. With this variable ORDO2 we can easily prescribe a missing value to LAG2_X for second observation in each ORDER group, just by writing expression 'if ordo2=2 then lag2_x=. ;'.

```
data laggi4;
  set one_by;
  by ordo;
  if first.ordo then ordo2=0;
   ordo2+1;
   lag1_x=lag(x);
   lag2_x=lag2(x);
   if first.ordo then
   do;
       lag1_x=. ;
```

```
      lag2_x=. ;
   end;
   if ordo2=2 then lag2_x=.;
run;
```

Table 5. LAG1 and LAG2 with two small tricks in the stratified data set

Obs	x	ordo	ordo2	lag1_x	lag2_x
1	75	0	1	.	.
2	32	0	2	75	.
3	18	0	3	32	75
4	91	0	4	18	32
5	36	0	5	91	18
6	22	0	6	36	91
7	79	1	1	.	.
8	40	1	2	79	.
9	12	1	3	40	79
10	19	1	4	12	40
11	78	1	5	19	12
12	44	1	6	78	19
13	97	1	7	44	78
14	26	2	1	.	.
15	71	2	2	26	.
16	55	2	3	71	26
17	53	2	4	55	71
18	86	2	5	53	55
19	14	2	6	86	53
…….					

In case we need the values of LAG3, LAG4, LAG5...LAGk we just add the relevant expressions like

```
if ordo2=3 then lag3_x=.;
if ordo2=4 then lag4_4=.;
.....
if ordo2=K then lagK=.;
```

And again, even more easy we can get the same result using IFN function instead of IF...THEN ...

This time in addition to the statement ".. lag1_x = ifn(first.ordo, ., lag(x));" we have to use "..lag2_x = ifn(first.ordo or ordo2=2, ., lag2(x));".
In case we need the values of LAG3, LAG4, LAG5..LAGk we just add the relevant expressions like:

```
lag3_x = ifn(first.ordo or ordo2=3, ., lag3(x));
lag4_x = ifn(first.ordo or ordo2=4, ., lag4(x));
  ......
lagK_x = ifn(first.ordo or ordo2=K, ., lagK(x));
```

For example,

```
data laggi4a;
  set one_by;
  by ordo;
  if first.ordo then ordo2=0;
  ordo2+1;
  lag1_x = ifn(first.ordo, ., lag(x));
  lag2_x = ifn( first.ordo or ordo2=2, ., lag2(x));
run;
```

USING QUASI LAG1 AND QUASI LAG2 WITHOUT LAG FUNCTION AND ADDITIONAL ORDERING VARIABLES

The approaches described above are effective but below we would like to suggest another one, which is even more efficient and uses neither LAG function, nor additional ordering variables, nor even IFN function. This approach is based on the modification of automatic variable *_N_*, creating a separate data set for each new *_N_* and then merging these data sets together. It can bring to each observation the values from as many as needed preceding records belonging to the same ORDER_BY group only.

Below we present an example of code, which successfully brings to each observations the values of variable X from two preceding observations, the same way as LAG1 and LAG2 functions would do.

The variable PIK serves to keep the values of automatic variable *_N_*:

```
data one_by;
   set one_by;
   pik=_n_;
run;
```

By adding 1 or 2 to PIK we create variables PIK1 and PIK2 in data sets PIKO1 and PIKO2 which are the copies of the original ONE_BY with values of PIK shifted by one or two observations, respectively. Then we rename PIK1 and PIK2 back to PIK. For the clarity purpose only, we are renaming variable X in the data set PIKO1 into LAG1_X and in the PIKO2 into LAG2_X (again, we are not using LAG function here):

```
data piko1(keep=pik1 x ordo rename=(pik1=pik x=lag1_x)) pik2(keep=pik2 x ordo
rename=(pik2=pik x= lag2_x))  ;
  set one_by;
  pik1=pik+1;
  pik2 =pik+2;
run;
```

At the next step, we merge all three data sets by ORDO and PIK taking only observations which have ORDO and PIK values from the data set ONE_BY (statement "if a"does this job). First, it means that merging will take place inside each ORDO group only. Second, inside each ORDO group, the first observation will have missing values of LAG1_X because PIKO1 did not have the relevant value of PIK (which is 1), and the same is true for LAG2_X, because PIKO2 did not have the relevant value of PIK (which is 2). It is obvious with the

PIK=1 in ONE_BY data set. The PIKO1 and PIKO2 data sets just do not have PIK=1 or PIK=2, because their PIK values are result of the adding 1 or 2 to PIK of the ONE_BY.

```
data fin;
  merge one_by(in=a) piko1 piko2 ;
  by ordo pik;
  if a;
run;
```

Table 5. Quasi LAG1 and Quasi LAG2 without using LAG function and additional ordering variable in the stratified data set

Obs	x	ordo	pik	lag1_x	lag2_ x
1	75	0	1	.	.
2	32	0	2	75	.
3	18	0	3	32	75
4	91	0	4	18	32
5	36	0	5	91	18
6	22	0	6	36	91
7	79	1	7	.	.
8	40	1	8	79	.
9	12	1	9	40	79
10	19	1	10	12	40
11	78	1	11	19	12
12	44	1	12	78	19
13	97	1	13	44	78
14	26	2	14	.	.
15	71	2	15	26	.
16	55	2	16	71	26
17	53	2	17	55	71
18	86	2	18	53	55
19	14	2	19	86	53
20	86	2	20	14	86

In order to see in details how the merge of ONE_BY, PIKO1 and PIKO2 gives us the result same as LAG function does, let us compare these three data sets and data set FIN:

Data set ONE_BY

Obs	x	ordo	pik
1	75	0	1
2	32	0	2
3	18	0	3
4	91	0	4
5	36	0	5
6	22	0	6
7	79	1	7
8	40	1	8
9	12	1	9
10	19	1	10
11	78	1	11
12	44	1	12

Data set PIKO1

Obs	x1	ordo	pik
1	75	0	2
2	32	0	3
3	18	0	4
4	91	0	5
5	36	0	6
6	22	0	7
7	79	1	8
8	40	1	9
9	12	1	10
10	19	1	11
11	78	1	12
12	44	1	13

Data set PIKO2

Obs	x1	ordo	pik
1	75	0	3
2	32	0	4
3	18	0	5
4	91	0	6
5	36	0	7
6	22	0	8
7	79	1	9
8	40	1	10
9	12	1	11
10	19	1	12
11	78	1	13
12	44	1	14

Data set FIN

Obs	x	ordo	pik	lag1_x	lag2_ x
1	75	0	1	.	.
2	32	0	2	75	.
3	18	0	3	32	75
4	91	0	4	18	32
5	36	0	5	91	18
6	22	0	6	36	91
7	79	1	7	.	.
8	40	1	8	79	.
9	12	1	9	40	79
10	19	1	10	12	40
11	78	1	11	19	12
12	44	1	12	78	19

In observation #1 of data set FIN both, LAG1_X and LAG2_X have missing values because in ORDO=0 group there is no PIK=1 either in PIKO1 or PIKO2 data set. In observation #2 of data set FIN we have LAG1_X=75, which comes from PIKO1, observation #2, PIK=2, and LAG2_X has missing value because there is no PIK=2 in PIKO2 data set (the source of values for LAG2_X variable). In observation #3 of data set FIN we have LAG1_X=32, which comes from PIKO1, observation #2, PIK=3, and LAG2_X=75, which comes from PIKO2 data set, observation #1, PIK=3. A similar logic is applicable to the rest of ORDO=0 group. At the beginning of ORDO=1 group we start over - in observation #8 of data set FIN both, LAG1_X and LAG2_X have missing values because in ORDO=1 group there is no PIK=7 either in PIKO1 or PIKO2 data set. And so on…

CONCLUSION

The suggested approach based on the modification of automatic variable _n_, creating a separate data set for each new _n_ and merging these data sets together is simple and effective. It does not use neither LAG function, nor additional ordering variables and allows the programmer to combine the values from as many as needed preceding records belonging to the same BY-group.

REFERENCES

1. Yunchao, Tian (2009). LAG – the Very Powerful and Easily Misused SAS ® Function. SAS Global Forum, Paper 055-2009.

2. Anjan Matlapudi and J. Daniel Knapp (2010). Please Don't Lag Behind LAG! NESUG 2010, Coder's Coner.

CONTACT INFORMATION

Alexander M Feigin, Alex MedStat, Inc.
Work Phone: 267-226-7772
Email:feiginy@aol.com