By-Group and Nearest Neighbor Processing Using PROC SQL

Kirk Paul Lafler, @sasNerd, Spring Valley, California

Introduction

A technique that SAS programmers often need and find useful is the ability to identify the beginning and ending observation in a by-group. The DATA step creates two temporary variables for each BY variable: FIRST.variable and LAST.variable. Performing by-group processing to identify FIRST., LAST., and BETWEEN observations is a popular technique with SAS users. Unfortunately, PROC SQL users had no way to emulate this very important programming technique. So, after considerable research I came up with coding techniques that can be used to emulate these stalwart DATA step programming techniques using subqueries. The SQL procedure can be used to correctly emulate and identify FIRST, LAST, and BETWEEN rows in By-Groups, and LAG and LEAD nearest neighbor processing.

Example Tables

The examples used in this paper utilize a database of two tables. (A relational database is a collection of tables.) The data used in all the examples in this paper consists of a selection of movies that I've viewed over the years. The Movies table consists of six columns: title, length, category, year, studio, and rating. Title, category, studio, and rating are defined as character columns with length and year being defined as numeric columns. The Actors table consists of three columns: title, actor_leading, and actor_supporting, all of which are defined as character columns. The two tables are illustrated below.

MOVIES Table

	Title	Length	Category	Year	Studio	Rating
1	Brave Heart	177	Action Adventure	1995	Paramount Pictures	R
2	Casablanca	103	Drama	1942	MGM / UA	PG
3	Christmas Vacation	97	Comedy	1989	Warner Brothers	PG-13
4	Coming to America	116	Comedy	1988	Paramount Pictures	R
5	Dracula	130	Horror	1993	Columbia TriStar	R
6	Dressed to Kill	105	Drama Mysteries	1980	Filmways Pictures	R
7	Forrest Gump	142	Drama	1994	Paramount Pictures	PG-13
8	Ghost	127	Drama Romance	1990	Paramount Pictures	PG-13
9	Jaws	125	Action Adventure	1975	Universal Studios	PG
10	Jurassic Park	127	Action	1993	Universal Pictures	PG-13
11	Lethal Weapon	110	Action Cops & Robber	1987	Warner Brothers	R
12	Michael	106	Drama	1997	Warner Brothers	PG-13
13	National Lampoon's Vacation	98	Comedy	1983	Warner Brothers	PG-13
14	Poltergeist	115	Horror	1982	MGM / UA	PG
15	Rocky	120	Action Adventure	1976	MGM / UA	PG
16	Scarface	170	Action Cops & Robber	1983	Universal Studios	R
17	Silence of the Lambs	118	Drama Suspense	1991	Orion	R
18	Star Wars	124	Action Sci-Fi	1977	Lucas Film Ltd	PG
19	The Hunt for Red October	135	Action Adventure	1989	Paramount Pictures	PG
20	The Terminator	108	Action Sci-Fi	1984	Live Entertainment	R
21	The Wizard of Oz	101	Adventure	1939	MGM / UA	G
22	Titanic	194	Drama Romance	1997	Paramount Pictures	PG-13

ACTORS Table

	Title	Actor_Leading	Actor_Supporting
1	Brave Heart	Mel Gibson	Sophie Marceau
2	Christmas Vacation	Chevy Chase	Beverly D'Angelo
3	Coming to America	Eddie Murphy	Arsenio Hall
4	Forrest Gump	Tom Hanks	Sally Field
5	Ghost	Patrick Swayze	Demi Moore
6	Lethal Weapon	Mel Gibson	Danny Glover
7	Michael	John Travolta	Andie MacDowell
8	National Lampoon's Vacation	Chevy Chase	Beverly D'Angelo
9	Rocky	Sylvester Stallone	Talia Shire
10	Silence of the Lambs	Anthony Hopkins	Jodie Foster
11	The Hunt for Red October	Sean Connery	Alec Baldwin
12	The Terminator	Arnold Schwarzenegge	Michael Biehn
13	Titanic	Leonardo DiCaprio	Kate Winslet

Identifying FIRST Rows in By-groups

SQL Code

```
/*****************************************************/
/** ROUTINE.....: FIRST-BY-GROUP-ROWS               **/
/** PURPOSE.....: Derive the first (min) row within **/
/**               each by-group using a subquery.   **/
/*****************************************************/
proc sql;
  create table first_bygroup_rows as
    select rating,
           title,
           'FirstRow' as ByGroup
      from movies M1
        where title =
              (select min(title)
                 from movies M2
                   where M1.rating = M2.rating)
          order by rating, title;
```

FIRST Rows Results

Rating	Title	ByGroup
G	The Wizard of Oz	FirstRow
PG	Casablanca	FirstRow
PG-13	Christmas Vacation	FirstRow
R	Brave Heart	FirstRow

Identifying LAST Rows in By-groups

SQL Code

```
/*****************************************************/
/** ROUTINE.....: LAST-BY-GROUP-ROWS                **/
/** PURPOSE.....: Derive the last (max) row within  **/
/**               each by-group using a subquery.   **/
/*****************************************************/
  create table last_bygroup_rows as
    select rating,
           title,
           'LastRow' as ByGroup
      from movies M1
        where title =
              (select max(title)
                 from movies M2
                   where M1.rating = M2.rating)
          order by rating, title;
```

LAST Rows Results

Rating	Title	ByGroup
G	The Wizard of Oz	LastRow
PG	The Hunt for Red October	LastRow
PG-13	Titanic	LastRow
R	The Terminator	LastRow

Identifying BETWEEN Rows in By-groups

SQL Code

```
/*********************************************************/
/** ROUTINE.....: BETWEEN-BY-GROUP-ROWS                 **/
/** PURPOSE.....: Derive not the first (min) row and not  **/
/**               the last (max) row within each By-group. **/
/*********************************************************/
 create table between_bygroup_rows as
    select rating,
           title,
           min(title) as Min_Title,
           max(title) as Max_Title,
           'BetweenRow' as ByGroup
      from movies
        group by rating
          having CALCULATED min_Title NOT = CALCULATED max_Title
             AND CALCULATED min_Title NOT = Title
             AND CALCULATED max_Title NOT = Title
           order by rating, title;
```

BETWEEN Rows Results

Rating	Title	Min_Title	Max_Title	ByGroup
PG	Jaws	Casablanca	The Hunt for Red October	BetweenRow
PG	Poltergeist	Casablanca	The Hunt for Red October	BetweenRow
PG	Rocky	Casablanca	The Hunt for Red October	BetweenRow
PG	Star Wars	Casablanca	The Hunt for Red October	BetweenRow
PG-13	Forrest Gump	Christmas Vacation	Titanic	BetweenRow
PG-13	Ghost	Christmas Vacation	Titanic	BetweenRow
PG-13	Jurassic Park	Christmas Vacation	Titanic	BetweenRow
PG-13	Michael	Christmas Vacation	Titanic	BetweenRow
PG-13	National Lampoon's Vacation	Christmas Vacation	Titanic	BetweenRow
R	Coming to America	Brave Heart	The Terminator	BetweenRow
R	Dracula	Brave Heart	The Terminator	BetweenRow
R	Dressed to Kill	Brave Heart	The Terminator	BetweenRow
R	Lethal Weapon	Brave Heart	The Terminator	BetweenRow
R	Scarface	Brave Heart	The Terminator	BetweenRow
R	Silence of the Lambs	Brave Heart	The Terminator	BetweenRow

SQL Code

```
/*************************************************************/
/** ROUTINE.....: CONCATENATE-FIRST-BETWEEN-LAST            **/
/** PURPOSE.....: Concatenate the results from the first    **/
/**                (min) row, between rows, and last (max)  **/
/**                row within each by-group, and print.     **/
/*************************************************************/
  create table first_between_last_rows as
    select rating, title, bygroup
      from first_bygroup_rows
    UNION ALL
    select rating, title, bygroup
      from between_bygroup_rows
    UNION ALL
    select rating, title, bygroup
      from last_bygroup_rows;
  select * from first_between_last_rows;
quit;
```

FIRST, LAST and BETWEEN Row Results

Row	Rating	Title	ByGroup
1	G	The Wizard of Oz	FirstRow
2	PG	Casablanca	FirstRow
3	PG-13	Christmas Vacation	FirstRow
4	R	Brave Heart	FirstRow
5	PG	Jaws	BetweenRow
6	PG	Poltergeist	BetweenRow
7	PG	Rocky	BetweenRow
8	PG	Star Wars	BetweenRow
9	PG-13	Forrest Gump	BetweenRow
10	PG-13	Ghost	BetweenRow
11	PG-13	Jurassic Park	BetweenRow
12	PG-13	Michael	BetweenRow
13	PG-13	National Lampoon's Vacation	BetweenRow
14	R	Coming to America	BetweenRow
15	R	Dracula	BetweenRow
16	R	Dressed to Kill	BetweenRow
17	R	Lethal Weapon	BetweenRow
18	R	Scarface	BetweenRow
19	R	Silence of the Lambs	BetweenRow
20	G	The Wizard of Oz	LastRow
21	PG	The Hunt for Red October	LastRow
22	PG-13	Titanic	LastRow
23	R	The Terminator	LastRow

Performing Nearest Neighbor Processing by Emulating LAG and LEAD Functionality

As a general rule SQL queries are designed to perform operations on a row-by-row basis. But, sometimes a problem comes along where row-by-row processing will not work. For example, nearest neighbor problems identify content that is close to another value. In the DATA step, LAG and LEAD functions are often used to perform these types of operations. But, PROC SQL does not support the use of the LAG and LEAD functions, and produces ERROR messages as is shown, below.

SQL Code

```
PROC SQL NONUMBER ;
  SELECT Title, Rating,
         LAG(Title)  AS Lag_Title,
         LEAD(Title) AS Lead_Title
    FROM Movies ;
QUIT ;
```

SQL Log

```
PROC SQL NONUMBER ;
  SELECT Title, Rating,
         LAG(Title)  AS Lag_Title,
         LEAD(Title) AS Lead_Title
    FROM Movies ;
ERROR: The LAG function is not supported in PROC SQL, it is only valid
       within the DATA step.
ERROR: Function LEAD could not be located.
QUIT ;
```

To overcome the unavailability of the LAG and LEAD functions in PROC SQL, the following code emulates LAG and LEAD functionality.

SQL Code

```
ODS OUTPUT SQL_RESULTS=Movies_with_Row_Numbers ;

PROC SQL NUMBER NOPRINT ;
 SELECT title, rating FROM mydata.Movies ;
QUIT ;

PROC SQL NONUMBER ;
 CREATE TABLE Nearest_Neighbor(drop=Row) as
   SELECT *,
     (SELECT Title
        FROM Movies_with_Row_Numbers
          WHERE Row = M.Row - 1) AS Previous_Title,
     (SELECT Title
        FROM Movies_with_Row_Numbers
          WHERE Row = M.Row + 1) AS Next_Title
     FROM Movies_with_Row_Numbers M ;
 SELECT * FROM Nearest_Neighbor ;
QUIT ;
```

Results

Title	Rating	Previous_Title	Next_Title
Brave Heart	R		Casablanca
Casablanca	PG	Brave Heart	Christmas Vacation
Christmas Vacation	PG-13	Casablanca	Coming to America
Coming to America	R	Christmas Vacation	Dracula
Dracula	R	Coming to America	Dressed to Kill
Dressed to Kill	R	Dracula	Forrest Gump
Forrest Gump	PG-13	Dressed to Kill	Ghost
Ghost	PG-13	Forrest Gump	Jaws
Jaws	PG	Ghost	Jurassic Park
Jurassic Park	PG-13	Jaws	Lethal Weapon
Lethal Weapon	R	Jurassic Park	Michael
Michael	PG-13	Lethal Weapon	National Lampoon's Vacation
National Lampoon's Vacation	PG-13	Michael	Poltergeist
Poltergeist	PG	National Lampoon's Vacation	Rocky
Rocky	PG	Poltergeist	Scarface
Scarface	R	Rocky	Silence of the Lambs
Silence of the Lambs	R	Scarface	Star Wars
Star Wars	PG	Silence of the Lambs	The Hunt for Red October
The Hunt for Red October	PG	Star Wars	The Terminator
The Terminator	R	The Hunt for Red October	The Wizard of Oz
The Wizard of Oz	G	The Terminator	Titanic
Titanic	PG-13	The Wizard of Oz	

Conclusion

The SQL procedure is a wonderful tool for SAS users to explore and use in a variety of application situations. This paper has presented code that emulates and correctly identifies the FIRST, LAST, and BETWEEN rows in By-Groups using and LAG and LEAD nearest neighbor processing using SAS-SQL (PROC SQL).

References

Lafler, Kirk Paul (2019). *PROC SQL: Beyond the Basics Using SAS, Third Edition*, SAS Institute Inc., Cary, NC, USA.

Lafler, Kirk Paul (2015), *"Five Little Known, But Highly Valuable and Widely Usable, PROC SQL Programming Techniques,"* Ohio SAS Users Group (OhioSUG) 2015 Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2014), *"Five Little Known, But Highly Valuable and Widely Usable, PROC SQL Programming Techniques,"* Wisconsin-Illinois SAS Users Group (WIILSUG) 2014 Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2013), *"Quick Results with PROC SQL,"* North East SAS Users Group (NESUG) 2013 Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2011), *"Quick Results with PROC SQL,"* Western Users of SAS Software (WUSS) 2011 Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2011), *"Quick Results with PROC SQL,"* MidWest SAS Users Group (MWSUG) 2011 Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2011), *"Powerful and Sometimes Hard-to-find PROC SQL Features,"* PharmaSUG 2011 Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2010), *"Exploring DATA Step Merges and PROC SQL Joins,"* South Central SAS Users Group (SCSUG) 2010 Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2009), *"Exploring DICTIONARY Tables and SASHELP Views,"* South Central SAS Users Group (SCSUG) Conference (November 8th – November 10th, 2009), Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2009), *"Exploring DICTIONARY Tables and SASHELP Views,"* Western Users of SAS Software (WUSS) Conference (September 1st – September 4th, 2009), Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2009), *"Exploring DICTIONARY Tables and SASHELP Views,"* PharmaSUG SAS Users Group Conference (May 31st – June 3rd, 2009), Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2008), *"Kirk's Top Ten Best PROC SQL Tips and Techniques,"* Wisconsin Illinois SAS Users Conference (June 26th, 2008), Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2008), *"Undocumented and Hard-to-find PROC SQL Features,"* Greater Atlanta SAS Users Group (GASUG) Meeting (June 11th, 2008), Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2008), *"Undocumented and Hard-to-find PROC SQL Features,"* PharmaSUG SAS Users Group Conference (June 1st - 4th, 2008), Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2008), *"Undocumented and Hard-to-find PROC SQL Features,"* Michigan SAS Users Group (MSUG) Meeting (May 29th, 2008), Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2008), *"Undocumented and Hard-to-find PROC SQL Features,"* Vancouver SAS Users Group Meeting (April 23rd, 2008), Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2008), *"Undocumented and Hard-to-find PROC SQL Features,"* Philadelphia SAS Users Group (PhilaSUG) Meeting (March 13th, 2008), Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2007), *"Undocumented and Hard-to-find PROC SQL Features,"* Proceedings of the PharmaSUG 2007 Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul and Ben Cochran (2007), *"A Hands-on Tour Inside the World of PROC SQL Features,"* Proceedings of the SAS Global Forum (SGF) 2007 Conference, Software Intelligence Corporation, Spring Valley, CA, and The Bedford Group, USA.

Lafler, Kirk Paul (2006), *"A Hands-on Tour Inside the World of PROC SQL,"* Proceedings of the Thirty-first Annual SAS Users Group International (SUGI) Conference.

Lafler, Kirk Paul (2005), *"A Hands-on Tour of the 5 Most Exciting Features Found in PROC SQL,"* Proceedings of the Thirteenth Annual Western Users of SAS Software Conference.

Lafler, Kirk Paul (2005), "*Manipulating Data with PROC SQL*," Proceedings of the Thirtieth Annual SAS Users Group International (SUGI) Conference.

Trademark Citations

Author Information

Kirk Paul Lafler is an entrepreneur, consultant, educator and author, and has used SAS software since 1979. Currently, Kirk works at San Diego State University as a lecturer and adjunct professor; at the University of California San Diego Extension as an advisor and adjunct professor; and teaches SAS, SQL, Python and R courses, seminars, workshops and webinars to users around the world. As the author of PROC SQL: Beyond the Basics Using SAS, Third Edition (SAS Press. 2019), Google® Search Complete (Odyssey Press. 2014) and hundreds of SAS papers and articles; Kirk has served as an Invited speaker, educator, keynote and section leader at SAS user group conferences and meetings worldwide; and is the recipient of 25 "Best" contributed paper, hands-on workshop (HOW), and poster awards.

Comments and suggestions can be sent to:

Kirk Paul Lafler
SAS® Consultant, Application Developer, Programmer, Data Analyst, Educator and Author
E-mail: KirkLafler@cs.com
LinkedIn: https://www.linkedin.com/in/KirkPaulLafler/
LinkedIn: https://www.linkedin.com/in/Order-of-Magnitude-Analytics/
Twitter: @sasNerd