

POWER SAS: A Survival Guide

Copyright ©2002 by Kirk Paul Lafler

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

Library of Congress Cataloging-in-Publication Data

Lafler, Kirk Paul, 1956-

Power SAS : a survival guide / Kirk Paul Lafler.  
p. cm.

ISBN 1-59059-066-X

1. SAS (Computer file) I. Title.

QA276.4 .L34 2002

005.3'042--dc21

2002013258

Printed and bound in the United States of  
America 12345678910

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The information in this book is distributed on an "as is" basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

Technical Reviewers: Charles Shipp,  
Mike DiGregorio, and Sunil Kumar Gupta

Editorial Directors: Dan Appleman, Gary Cornell,  
Jason Gilmore, Simon Hayes, Karen Watterson,  
John Zukowski

Managing Editor: Grace Wong

Project Manager and Development Editor:  
Tracy Brown Collins

Copy Editor: Corbin Collins

Production Manager: Kari Brooks

Compositor: Diana Van Winkle,  
Van Winkle Design

Artist: Cara Brunk, Blue Mud Productions

Indexer: Rebecca Plunkett

Cover Designer: Kurt Krames

Manufacturing Manager: Tom Debolski

Marketing Manager: Stephanie Rodriguez

Distributed to the book trade in the United States  
by Springer-Verlag New York, Inc., 175 Fifth  
Avenue, New York, NY, 10010 and outside the  
United States by Springer-Verlag GmbH & Co. KG,  
Tiergartenstr. 17, 69112 Heidelberg, Germany.  
In the United States, phone 1-800-SPRINGER,  
email [orders@springer-ny.com](mailto:orders@springer-ny.com), or visit  
<http://www.springer-ny.com>.  
Outside the United States, fax +49 6221 345229,  
email [orders@springer.de](mailto:orders@springer.de), or visit  
<http://www.springer.de>.

For information on translations, please contact  
Apress directly at 2560 9th Street, Suite 219,  
Berkeley, CA 94710. Phone 510-549-5930,  
fax 510-549-5939, email [info@apress.com](mailto:info@apress.com),  
or visit <http://www.apress.com>.










The source code for this book is available  
to readers at <http://www.apress.com> in the  
Downloads section. You will need to answer  
questions pertaining to this book in order to  
successfully download the code.





## CHAPTER 4

# Data Manipulation

THE ABILITY TO manipulate data in the Base SAS software plays a very important role in the preparation of data for data analysis. This chapter presents tips directly related to changing the way your data looks without changing the data itself. You'll learn essential techniques on the different ways of subsetting data, reshaping columns of data, and sending output to SAS data sets.

In this chapter, you'll learn how to

-  Create new SAS data sets with a subsetting IF and WHERE
-  Use SAS operators to subset data
-  Use SAS statements such as IF-THEN/ELSE, DELETE, and SELECT to subset data
-  Use System options to subset data
-  Reshape columns of data with KEEP= / DROP= data set options
-  Reshape data with a SET statement
-  Reshape data by using the APPEND procedure
-  Reshape data with multiple SET statements
-  Reshape data by interleaving SAS data sets

-  Reshape data by merging two or more SAS data sets
-  Reshape data by joining in the SQL procedure
-  Send output results to a SAS data set
-  Use a variety of summary procedures to send output to a SAS data set

## Subsetting Data

This section includes tips on how to subset data either by restricting the number of observations, variables, or both. You'll learn how to subset your data quickly and easily with these handy data manipulation tips.

### 1 Creating new data sets with a subset

New SAS data sets can be created from one or more external files or existing SAS data sets. When a data set is a subset, it contains part or all of the records or observations from the original external file or data set. Subsets generally contain fewer records or observations than the original external file or SAS data set they are derived from.



The `TITLE` and `RATING` variables from the `PG13_RATING` data set were renamed to prevent the data from being overwritten.



It's best to interleave data sets with similar data structures to prevent missing values from being assigned.

## 16 Reshaping data with three SET statements

When three **SET** statements are specified in a **DATA** step, the objective is to pair the observations from each data set. The pairing of observations continues until one or more of the data sets run out of observations. In the next example, three SET statements are specified to join “PG,” “PG-13,” and “R” movies together.



If similar variables exist in any of the data sets, each variable needs to be renamed to prevent the data from being overwritten.

Code:

```
DATA SET_JOIN;
  SET PG_RATING(KEEP=TITLE RATING
                RENAME=(TITLE=PG_TITLE
                        RATING=PG_RATING));
  SET PG13_RATING(KEEP=TITLE RATING
                 RENAME=(TITLE=PG13_TITLE
                         RATING=PG13_RATING));
  SET R_RATING(KEEP=TITLE RATING
              RENAME=(TITLE=R_TITLE
                      RATING=R_RATING));
RUN;
```

## 17 Interleaving SAS data sets

The process of interleaving SAS data sets involves concatenating observations in sorted order. Before the data can be interleaved, it must first be verified that it is ordered by the **BY** variable. In the next example, “PG” and “PG-13” movies are interleaved in order of their length.

Code:

```
PROC SORT DATA=PG_RATING;
  BY LENGTH;
RUN;
PROC SORT DATA=PG13_RATING;
  BY LENGTH;
RUN;
DATA INTERLEAVE;
  SET PG_RATING(KEEP=TITLE LENGTH RATING)
    PG13_RATING(KEEP=TITLE LENGTH RATING);
  BY LENGTH;
RUN;
PROC PRINT NOOBS;
RUN;
```

Results:










Title	Length	Rating
Christmas Vacation	97	PG-13
National Lampoon's Vacation	98	PG-13
Casablanca	103	PG
Michael	106	PG-13
Poltergeist	115	PG
Rocky	120	PG
Star Wars	124	PG
Jaws	125	PG
Ghost	127	PG-13
Jurassic Park	127	PG-13
The Hunt for Red October	135	PG
Forrest Gump	142	PG-13
Titanic	194	PG-13







## CHAPTER 6

# Data Presentation

DATA PRESENTATION is a cornerstone of the Base SAS software. Reports display information in one of two ways: detail or summary form. SAS software provides numerous easy-to-use procedures to present the information you need, any time you need it. For a more custom look, you can create output presentations using the DATA step. This chapter presents a variety of tips on report-writing procedures, customizing reports with the DATA step, data presentation using ODS, and delivering online output using Web browser software.

In this chapter, you'll learn how to

-  Take advantage of useful PROC PRINT reporting features
-  Control orientation and labeling of column headers in PROC PRINT
-  Suppress column headers in PROC PRINT
-  Create custom-presentations with the DATA step
-  Create two reports in a single DATA step
-  Create two-, three-, and four-column output in the DATA step
-  Format output with global ODS statements
-  Create Rich Text Format (RTF) output
-  Create PostScript output

-  Integrate ODS into the DATA step
-  Distribute content anywhere and anytime using ODS
-  Create HTML output as an integrated Web application
-  Combine output results using ODS
-  Create Portable Document Format (PDF) output
-  Create powerful drill-down applications using ODS

## Report Writing and PROC PRINT

This section offers useful techniques when using PROC PRINT to better control the way the PRINT procedure formats output.

### 1 Suppressing the observation number in PROC PRINT output

By default, PROC PRINT automatically produces an observation number of output. Sometimes this can be a handy feature, but can also clutter the appearance of the report. By specifying the NOOBS option in the PROC PRINT statement, the observation number is suppressed.

```

PUT @1 TITLE $30.
  @35 RATING $5.;
IF EOF THEN PUT 'FINISHED!';
RETURN;
H: PUT 'DETAIL REPORT';
RETURN;
RUN;

```

## 7 Creating two reports in a single DATA step

When reporting requirements call for the creation of two reports from the same data, sometimes it is possible to create them both in a single DATA step. Typically this approach works best when one report contains detail information and the second report represents summary information. The primary advantage for creating two reports in a single step is to conserve on I/O resources (for more information, see Chapter 7, “Efficiency and Performance”). For example, two separate FILE statements and subroutines are coded, one for detail lines and the second for summary lines, as shown here.

*Code:*

```

DATA _NULL_;
  SET MOVIES END=EOF;
  FILE DETAIL LINESLEFT=LEFT
              HEADER=H NOTITLES;
  IF LEFT < 3 THEN PUT _PAGE_;
  PUT @1 TITLE $30.
    @35 RATING $5.;
  CTR_MOVIES+1;
  IF EOF THEN LINK SUMMARY;
RETURN;
H: PUT @15 'DETAIL REPORT';
RETURN;
SUMMARY:
  FILE SUMMARY NOTITLES;
  PUT @15 'SUMMARY REPORT'

```

```

    ///@12 'MY FAVORITE MOVIES'
    ///@12 'TOTAL MOVIES: ' CTR_MOVIES 3.;
RETURN;
RUN;

```

## 8 Creating two-column output in the DATA step

There are times when requirements call for output to be displayed in two columns. To accomplish this, a **FILE N=PS** (pagesize) option is specified to give the SAS System access to the entire page at a time. Then, two **DO loops** are used to control the positioning of output down rows and across columns. Next, within the DO loops, a **SET** statement is specified to control when the DATA step processes observations on input. Finally, a **PUT \_PAGE\_** statement is used to control printing the two columns of output per page.

The following example shows SAS code that first removes observations with duplicate BY values. It then prepares a two-column report similar to a two-column phone book. The maximum number of rows (controlled by the first DO loop) is set at 4 and the maximum number of columns (controlled by the second DO loop) is 16. Once the inner and outer DO groups are satisfied, the page is released with the PUT \_PAGE\_ statement.

*Code:*

```

PROC SORT DATA=MOVIES(KEEP=RATING)
          OUT=MOVIES_SORTED
          NODUPKEY;
  BY RATING;
RUN;
TITLE;
DATA _NULL_;
  FILE PRINT N=PS;

```