

CHAPTER

25

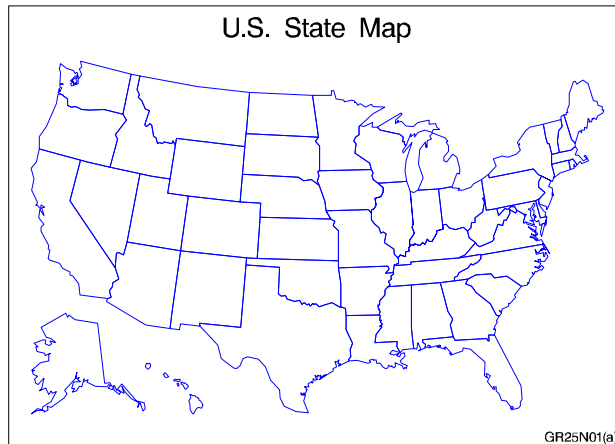
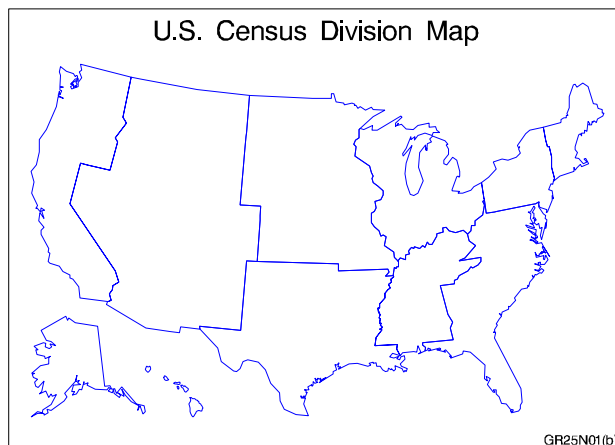
The GREMOVE Procedure

<i>Overview</i>	905
<i>Concepts</i>	906
<i>About the Input Map Data Set</i>	906
<i>About the Output Map Data Set</i>	907
<i>About Unmatched Area Boundaries</i>	907
<i>Procedure Syntax</i>	908
<i>PROC GREMOVE Statement</i>	908
<i>BY Statement</i>	909
<i>ID Statement</i>	910
<i>Examples</i>	910
<i>Example 1: Removing State Boundaries from U.S. Map</i>	910
<i>Example 2: Creating an Outline Map of Africa</i>	915

Overview

The GREMOVE procedure processes a map data set that is used as input. It does not produce any graphics output. Instead, it produces an output data set that typically becomes the input map data set for the GMAP procedure (see Chapter 19, “The GMAP Procedure,” on page 731).

The GREMOVE procedure combines unit areas defined in a map data set into larger unit areas by removing shared borders between the original unit areas. For example, Figure 25.1 on page 906 and Figure 25.2 on page 906 show combined unit areas in a typical map data set by removing state boundaries to create regional census divisions.

Figure 25.1 Map before Removing Borders (GR25N01(a))**Figure 25.2** Map after Removing Borders (GR25N01(b))

The program for these maps is shown in Example 1 on page 910.

Concepts

The GREMOVE procedure processes the input map data set to remove internal boundaries and creates a new map data set. The PROC GREMOVE statement identifies the input and output map data sets. The ID statement identifies the variable or variables in the input map data set that define the current unit areas. The BY statement identifies the variable or variables in the input map data set that define the new unit areas.

About the Input Map Data Set

The input map data set must be in map data set format (see “About Map Data Sets” on page 735) and it must contain these variables:

- a numeric variable named X that contains the horizontal coordinates of the map boundary points.
- a numeric variable named Y that contains the vertical coordinates of the map boundary points.
- one or more variables that uniquely identify the current unit areas in the map. These variables are listed in the ID statement. Each group of observations with a different ID variable value is evaluated as a separate unit area.
- one or more variables that identify the new unit areas to be created in the output map data set. These variables are listed in the BY statement.

It may also contain the variable SEGMENT, which is used to distinguish non-conterminous segments of the same unit areas. Other variables may exist in the input map data set, but they do not affect the GREMOVE procedure and they will not be carried into the output map data set.

About the Output Map Data Set

The output map data set contains the newly defined unit areas. These new unit areas are created by removing all interior line segments from the original unit areas. All variables in the input map data set except X, Y, SEGMENT, and the variables listed in the BY statement are omitted from the output map data set.

The output map data set may contain missing X, Y coordinates to construct any polygons that have enclosed boundaries (like lakes or combined regions that have one or more hollow interior regions).

The SEGMENT variable in the output map data set is ordered according to the size of the bounding box around the polygon that it describes. A SEGMENT value of 1 describes the polygon whose bounding box is the largest in the unit area and so on. This information is useful for removing small polygons that clutter up maps.

All current unit areas with common BY-variable value(s) are combined into a single unit area in the output map data set. The new unit area contains

- all boundaries that are not shared, such as islands and lakes
- all boundaries that are shared by two different BY groups.

About Unmatched Area Boundaries

If you are using map data sets in which area boundaries do not match precisely (for example, if the boundaries were digitized with a different set of points), PROC GREMOVE will not be able to identify common boundaries properly, resulting in abnormalities in your output data set.

If the points in the area boundaries match up except for precision differences, before using PROC GREMOVE round each X and Y value in your map data set accordingly, using the DATA step function ROUND. See *SAS Language Reference: Dictionary* for information on the ROUND function.

For example, if you have a map data set named APPROX in which the horizontal and vertical coordinate values for interior boundaries of unit areas are exactly equal only to three decimal places, this DATA step creates a new map data set, EXACT, that is better suited for use with the GREMOVE procedure:

```
data exact;
  set approx;
  if x ne . then x=round(x,.001);
  if y ne . then y=round(y,.001);
run;
```

Procedure Syntax

Requirements: The BY and ID statements are required.

```
PROC GREMOVE <DATA=input-map-data-set>
  <OUT=output-map-data-set>;
BY <DESCENDING>variable-l
  <...<DESCENDING>variable-n>
  <NOTSORTED>;
ID variable(s);
```

PROC GREMOVE Statement

Identifies the input and output map data sets.

Requirements: An input map data set is required.

Syntax

```
PROC GREMOVE <DATA=input-map-data-set>
  <OUT=output-map-data-set>;
```

Options

DATA=*input-map-data-set*

specifies the map data set that is to be processed. By default, the procedure uses the most recently created SAS data set. The GREMOVE procedure expects the observations in the input map data set to be sorted in ascending order of the BY-variable values.

See also: “About the Input Map Data Set” on page 906 and “SAS Data Sets” on page 25

Featured in: Example 2 on page 915

OUT=*output-data-set*

names the new map data set, which contains the coordinates of the new unit areas created by the GREMOVE procedure. By default, the GREMOVE procedure names the new data set using the DATA*n* naming convention. That is, the procedure uses the name WORK.DATA*n*, where *n* is the next unused number in sequence. Thus, the first automatically named data set is DATA1, the second is DATA2, and so on.

See also: “About the Output Map Data Set” on page 907

Featured in: Example 2 on page 915

BY Statement

Lists the variable or variables that identify the new unit areas.

Requirements: At least one variable is required.

See also: “BY Statement” on page 177

Featured in: Example 1 on page 910

Syntax

```
BY <DESCENDING>variable-1
    <...<DESCENDING>variable-n>
    <NOTSORTED>;
```

Required Arguments

variable(s)

identifies one or more variables in the input map data set that define the new unit areas. *Variable(s)* can be either numeric or character.

The BY variables in the input map data set become the ID variables for the output map data set.

Options

DESCENDING

indicates that the input map data set is sorted in descending order. By default, the GREMOVE procedure expects all BY-variable values to appear in ascending order.

This option affects only the variable that immediately follows the option.

NOTSORTED

indicates that observations with the same BY-variable values are to be grouped as they are encountered without regard for whether the values are in alphabetical or numerical order. NOTSORTED can appear anywhere in the BY statement. It affects all of the variables that are specified in the statement. NOTSORTED overrides DESCENDING if both appear in the same BY statement.

Ordering Observations

To sort the input map data set, use the SORT procedure in base SAS, for example

```
/* arrange the observations in desired order */
proc sort data=mapdata out=mapsort;
    by state;
run;
/* remove the county boundaries */
proc gremove data=mapsort out=newmap;
    by state;
    id county;
run;
```

Notice that the GREMOVE procedure uses the same BY statement as the SORT procedure.

See the *SAS Procedures Guide* for further information on the SORT procedure.

Note: If an observation is encountered for which the BY-variable value is out of the proper order, the GREMOVE procedure stops and issues an error message. \triangle

ID Statement

Identifies the variable or variables that define the hierarchy of the current unit areas in the input map data set.

Requirements: At least one *id-variable* is required.

Featured in: Example 1 on page 910

Syntax

ID *id-variable(s)*;

Required Arguments

id-variable(s)

specifies one or more variables in the input map data set that identify the unit areas to be combined. These variables are not included in the output map data set.

Id-variable(s) can be either numeric or character.

See also: “About the Input Map Data Set” on page 906

Examples

The following examples illustrate major features of the GREMOVE procedure.

Example 1: Removing State Boundaries from U.S. Map

Procedure features:

BY statement

ID statement

Other features:

SORT procedure

MERGE procedure

LIBNAME statement

Sample library member: GR25N01

This example processes the MAPS.US map data set, supplied with SAS/GRAPH, to produce a new map data set containing boundaries for the U.S. Bureau of the Census divisions. Because the MAPS.US map data set does not contain a variable to identify any unit area other than states, this example creates a map data set that contains the census divisions and that can be processed with the GREMOVE procedure.

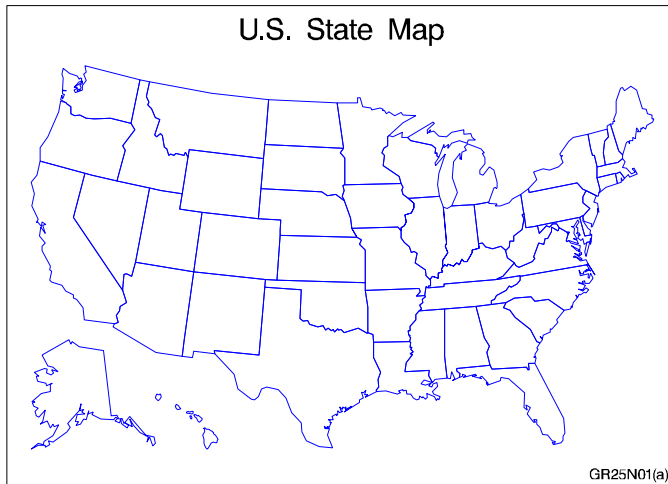
The STATE variable in the MAPS.US data set, containing numeric FIPS codes for each state, is used as the BY-variable to merge the REFLIB.CBSTATES and MAPS.US data sets. Output 25.1 on page 911 shows the variables that are present in the data set before using the GREMOVE procedure:

Output 25.1 The MAPS.US Data Set

OBS	STATE	MAPS.US Data Set		X	Y
		SEGMENT			
1	1	1		0.16175	-0.10044
2	1	1		0.12305	-0.10415
3	1	1		0.12296	-0.10678
.					
.					
1524	56	1		-0.18757	0.15035
1525	56	1		-0.10158	0.13997
1526	56	1		-0.10398	0.11343

And Figure 25.3 on page 912 shows the map before processing:

Figure 25.3 Map before Removing Borders (GR25N01(a))



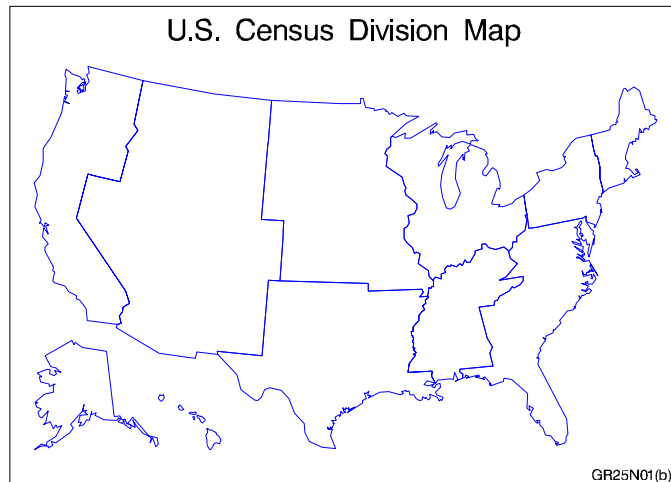
Output 25.2 on page 912 shows the variables that are present in the data set after you use the GREMOVE procedure. Notice that the new map data set contains a new variable called DIVISION:

Output 25.2 The REMSTATE Data Set

OBS	REMSTATE Data Set		SEGMENT	DIVISION
	X	Y		
1	0.29825	0.17418	1	1
2	0.29814	0.17820	1	1
3	0.30206	0.18045	1	1
.				
.				
.				
1082	-0.18715	-0.16010	8	9
1083	-0.18747	-0.15971	8	9
1084	-0.18747	-0.15951	8	9

Figure 25.4 on page 913 shows the new map after PROC GREMOVE has removed interior state boundaries.

Figure 25.4 Map after Removing Borders (GR25N01(a))



Assign the librefs and set the graphics environment. If the libref MAPS is already assigned, omit the LIBNAME statement.

```
libname reflib 'SAS-data-library';
libname maps 'SAS-maps-library';
goptions reset=global gunit=pct border cback=white
          colors=(black blue green red)
          ftext=swiss htitle=6 htext=3;
```

Create data set REFLIB.CBSTATES. This data set includes a variable, DIVISION, that contains the number of the U.S. Bureau of the Census division for the state. This data step converts letter codes to numeric FIPS codes that match those in the STATE variable of MAPS.US.

```
data reflib.cbstates;
  length state 8 stcode $ 2 division 4;
  input stcode division;
  state=stfips(stcode);
  drop stcode;
  datalines;
CT 1
MA 1
...more data lines...
OR 9
WA 9
;
```

Sort data set in FIPS-code order. Create a sorted data set, CBSORT. It can be properly match-merged with the MAPS.US map data set, which is already sorted in FIPS-code order.

```
proc sort data=reflib.cbstates out=cbsort;
  by state;
```

```
run;
```

Add DIVISION variable to map data set by merging the CBSORT data set with MAPS.US. Create a new map data set, USCB, that contains all of the state boundary coordinates from the MAPS.US data set plus the added variable DIVISION.

```
data uscb;
  merge cbsort maps.us;
  by state;
run;
```

Sort data set in DIVISION order. Sort USCB by the DIVISION variable to create the DIVSTATE data set.

```
proc sort data=uscb out=divstate;
  by division;
run;
```

Remove interior boundaries within divisions. BY specifies the variable, DIVISION, in the input map data set that identifies the new unit areas. ID specifies the variable, STATE, in the input map data set that identifies the current unit areas.

```
proc gremove data=divstate out=remstate;
  by division;
  id state;
run;
```

Define title and footnote for map.

```
title 'U.S. State Map';
footnote j=r 'GR25N01(a) ';
```

Define pattern characteristics.

```
pattern value=mempty repeat=48 color=blue;
```

Show the original map.

```
proc gmap map=maps.us data=maps.us all;
  id state;
  choro state / nolegend;
run;
```

Define new title and footnote for map.

```

title 'U.S. Census Division Map';
footnote j=r 'GR25N01(b) ';

```

Show the regional map. ID specifies the variable, DIVISION, that identifies the unit areas in the processed data set. CHORO specifies DIVISION as the response variable.

```

proc gmap map=remstate data=remstate all;
  id division;
  choro division / nolegend;
run;
quit;

```

Example 2: Creating an Outline Map of Africa

Procedure features:

PROC GREMOVE options:

DATA=

OUT=

Other features:

GMAP procedure

Sample library member: GR25N02

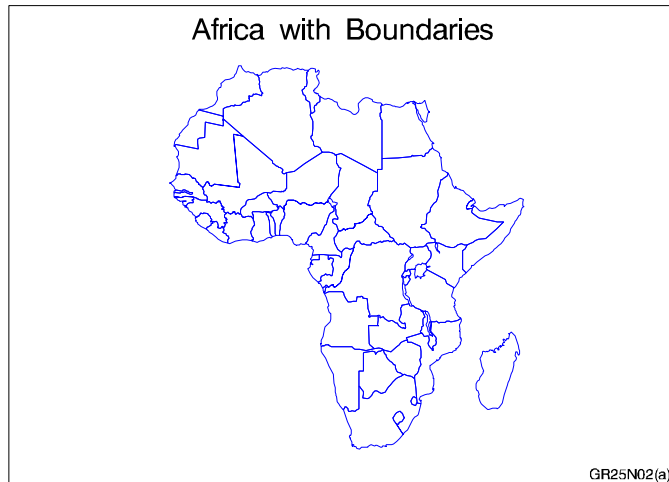
This example processes the MAPS.AFRICA map data set, supplied with SAS/GRAPH, to produce a new map data set that contains no internal boundaries. This is done by adding a new variable, REGION, to the map data set and setting it equal to 1. Unit areas from the input map data set that have the same BY-variable value are combined into one unit area in the output map data set. Output 25.3 on page 915 shows the variables present in the original map data set:

Output 25.3 The MAPS.AFRICA Data Set

OBS	MAPS.AFRICA Data Set			
	ID	SEGMENT	X	Y
1	125	1	0.57679	1.43730
2	125	1	0.57668	1.43467
3	125	1	0.58515	1.42363
.				
.				
.				
3462	990	1	1.04249	0.50398
3463	990	1	1.04184	0.50713
3464	990	1	1.04286	0.50841

Figure 25.5 on page 916 shows the map before processing:

Figure 25.5 Map before Removing Borders (GR25N02(a))



The new REFLIB.AFRICA map data set is created with a new variable, REGION. Output 25.4 on page 916 shows the variables that are present in the new map data set created by the GREMOVE procedure:

Output 25.4 The REFLIB.AFRICA Data Set

OBS	REFLIB.AFRICA Data Set			
	X	Y	SEGMENT	REGION
1	0.24826	1.02167	1	1
2	0.25707	1.02714	1	1
3	0.26553	1.03752	1	1
.				
.				
982	1.19071	1.30043	3	1
983	1.18675	1.30842	3	1
984	1.18518	1.32822	3	1

Figure 25.6 on page 917 shows the new map after PROC GREMOVE has removed all of the interior boundaries:

Figure 25.6 Map after Removing Borders (GR25N02(b))

**Assign the librefs and set the graphics environment.**

```
libname reflib 'SAS-data-library';
libname maps 'SAS-maps-library';
goptions reset=global gunit=pct border cback=white
          colors=(black blue green red)
          ftext=swiss htitle=6 htext=3;
```

Create the NEWAF data set. This new map data set contains all the variables in the SAS/GRAPH supplied MAPS.AFRICA map data set plus the added variable REGION.

```
data newaf;
  set maps.africa;
  region=1;
run;
```

Remove the unit areas from the AFRICA data set. DATA= specifies the input map data set and OUT= specifies the output map data set. The input map data set has a variable called REGION that is used as the BY-variable to identify the new unit areas. The ID statement specifies the current unit areas from the input map data set.

```
proc gremove data=newaf out=reflib.africa;
  by region;
  id id;
run;
```

Define the title and footnote.

```
title 'Africa with Boundaries';
footnote j=r 'GR25N02(a) ';
```

Define pattern characteristics.

```
pattern value=mempty r=50 color=blue;
```

Display the original map.

```
proc gmap map=maps.africa data=maps.africa all;  
  id id;  
  choro id / nolegend;  
run;
```

Define a new title and footnote for the map.

```
title 'Africa without Boundaries';  
footnote j=r 'GR25N02(b) ';
```

Display the map with no boundaries. ID specifies the variable, REGION, that identifies the unit areas in the processed data set.

```
proc gmap data=reflib.africa map=reflib.africa;  
  id region;  
  choro region / nolegend;  
run;  
quit;
```

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/GRAPH® Software: Reference, Version 8*, Cary, NC: SAS Institute Inc., 1999.

SAS/GRAPH® Software: Reference, Version 8

Copyright © 1999 by SAS Institute Inc., Cary, NC, USA.

ISBN 1-58025-525-6

All rights reserved. Printed in the United States of America.

U.S. Government Restricted Rights Notice. Use, duplication, or disclosure of the software by the government is subject to restrictions as set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

OS/2®, OS/390®, and IBM® are registered trademarks or trademarks of International Business Machines Corporation.

Other brand and product names are registered trademarks or trademarks of their respective companies.

The Institute is a private company devoted to the support and further development of its software and related services.