CHAPTER 23

The GPROJECT Procedure

Overview

The GPROJECT procedure processes map data sets by converting spherical coordinates (longitude and latitude) into Cartesian coordinates for use by the GMAP procedure. This process of converting coordinates from spherical to Cartesian is called *projecting*. In many of the geographic map data sets that are available with SAS/GRAPH software, the observation values are stored as longitude and latitude coordinates on a sphere (which means the map is unprojected). When these observations are plotted by the GMAP procedure, which is designed to plot points on a two-dimensional plane, the resulting map is often reversed and elongated as a result of forcing the curved map surface onto a flat plane.

The GPROJECT procedure enables you to use one of several map projection techniques to project the coordinates onto a two-dimensional plane while attempting to minimize the distortion of area, distance, direction, and shape properties of the original sphere. (The earth is not precisely spherical and the GPROJECT procedure does not attempt to correct this small distortion.) The output map data set that is produced by the procedure contains coordinate values expressed in Cartesian coordinates that can be displayed correctly using the GMAP procedure.

The GPROJECT procedure also can create a rectangular subset of the input map data set by excluding all points with longitude and latitude values that fall outside of a specified range. This provides a handy way to reduce the size of the map data set if you need only a portion of a larger map.

The GPROJECT procedure does not produce any graphics output. Instead, it produces an output map data set, which typically becomes the input map data set for the GMAP procedure (see Chapter 19, "The GMAP Procedure," on page 731).

Figure 23.1 on page 874 and Figure 23.2 on page 874 illustrate the effect of using GPROJECT defaults (Albers projection with standard parallels that are calculated by the procedure) to project a typical map data set with coordinates that are stored as longitude and latitude.

The program for the following maps can be seen in Example 1 on page 885.

Figure 23.1 Map before Projection (GR23N01(a))

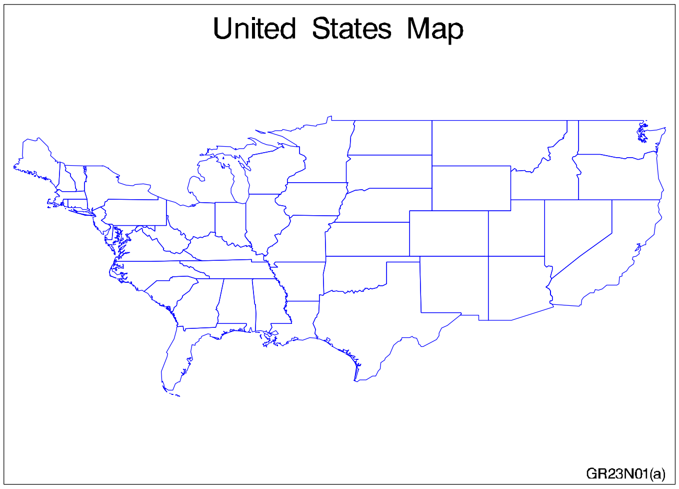
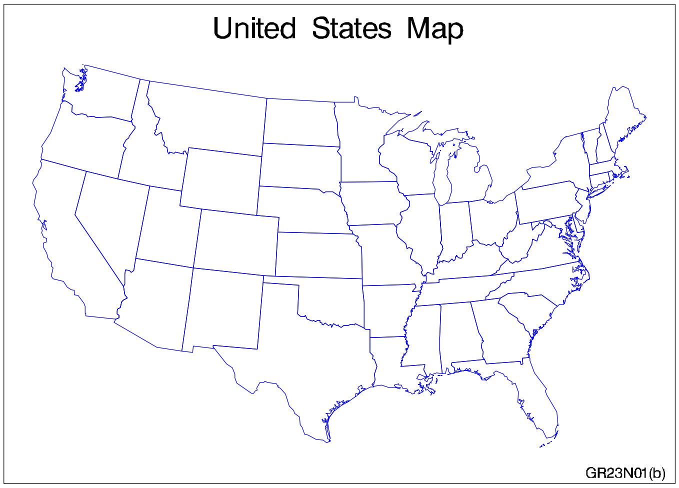


Figure 23.2 Map after Projection (GR23N01(b))



Concepts

About the Input Map Data Set

The input map data set must be in map data set format (see “SAS/GRAPH Map Data Sets” on page 761), and it must contain these variables:

□ a numeric variable named X that contains the longitude coordinates of the map boundary points.

□ a numeric variable named Y that contains the latitude coordinates of the map boundary points.

□ one or more *identification variables* that uniquely identify the unit areas in the map. These variables are listed in the ID statement.

The X and Y variables contain the values that are to be projected.

In addition, the input map data set also can contain

□ a numeric variable named SEGMENT that distinguishes nonconterminous segments of the unit areas.

□ a numeric variable named DENSITY that can be used to affect the output from PROC GPROJECT. See “Clipping Map Data Sets” on page 884 for more information.

Other variables in the input map data set do not affect the GPROJECT procedure.

Input Map Data Sets that Contain Only Unprojected Values

Note: Projection is appropriate for map data sets in which the X and Y variable values represent longitude and latitude. Some of the map data sets that are supplied with SAS/GRAPH have already been projected; such data set should not be projected again. △

The following is a list of all of the Institute-supplied data sets that contain X and Y variables whose values are unprojected:

CANADA3

CANADA4

COUNTIES

COUNTY

STATES

See Example 1 on page 885 for an illustration of this type of input map data set and the variables it contains.

Input Map Data Sets that Contain Both Projected and Unprojected Values

Most map data sets contain both sets of variables (X, Y and LONG, LAT) for projected and unprojected maps. In these cases, the X and Y variables will produce a projected map so you don’t need to use the GPROJECT procedure. However, you may want to use the LONG and LAT variables to reproject the map using a different projection type. To do this you must first rename the LONG and LAT variables. It is necessary to rename the LONG and LAT variables because the GPROJECT procedure looks for variables that are named X and Y by default. You can create a new map data set using the OUT= option, drop the current X and Y variables, and rename the LONG and LAT variables. Your new data set will then contain unprojected values in X and Y. The following statements illustrate how to do this:

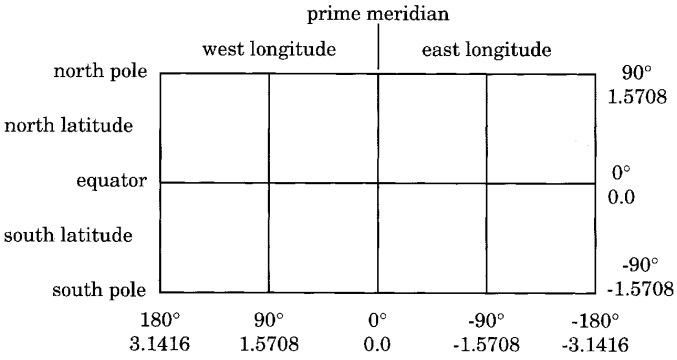
```
proc gproject data=map.austral
              (drop=x y rename=(long=x lat=y))
              out=reflib.newaust;
   id id;
run;
```

For additional information on the supplied SAS/GRAPH map data sets, see “SAS/GRAPH Map Data Sets” on page 761 and the METAMAPS data set in your maps data set directory.

About Coordinate Values

Figure 23.3 on page 876 shows the standard coordinate system for map data sets with coordinates in longitude and latitude. For the longitude and latitude values (below and to the right of the figure, respectively) the upper value is expressed in degrees and the lower value is expressed in radians. A radian is approximately 57.3 degrees.

Figure 23.3 Longitude and Latitude Coordinates



By default, the GPROJECT procedure assumes that the units for the input coordinate values are radians and that values for the horizontal coordinate increase from east to west across the map. If your map coordinates are stored as degrees of arc, use the DEGREE option in the PROC GPROJECT statement. If the horizontal coordinate values in the map increase west-to-east rather than east-to-west, use the EASTLONG option in the PROC GPROJECT statement. See “Options” on page 880 for details of DEGREE and EASTLONG.

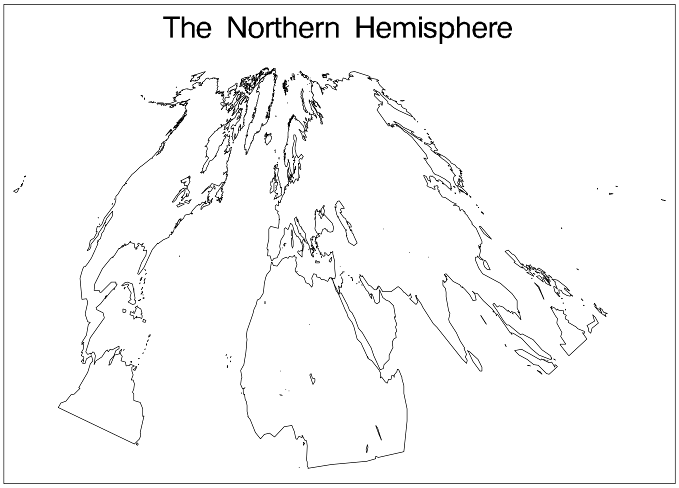
The unprojected map data sets that are provided with SAS/GRAPH can be projected if you use the default procedure characteristics: coordinate units in the data sets are radians, and horizontal values increase east-to-west.

About Types of Map Projections

The GPROJECT procedure performs three different types of projection: Albers’ equal-area projection with two standard parallels (the default method), Lambert’s conformal projection with two standard parallels, or the gnomonic projection (an azimuthal equidistant projection).

These sections describe the basic theory of each projection method. For comparison, Figure 23.4 on page 877 shows an unprojected map of the northern hemisphere.

Figure 23.4 Unprojected Map



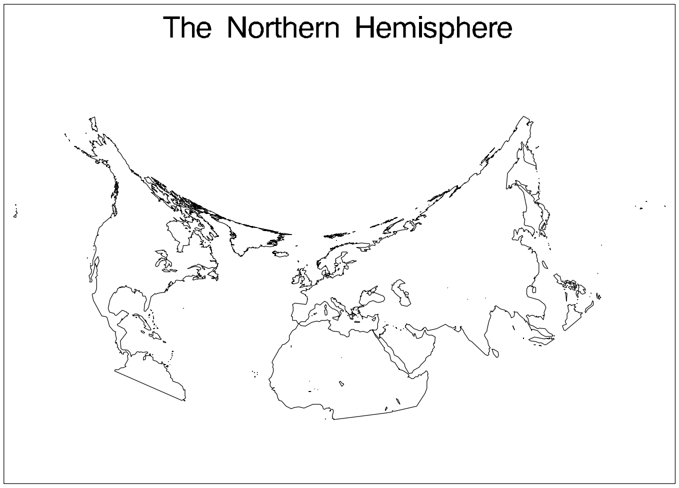
Albers' Equal-Area Projection

The *Albers' projection* is a conic projection from the surface of the sphere to a cone secant to the sphere, cutting it at two standard parallels of latitude. The axis of the cone coincides with an extension of the polar axis of the sphere. Each section of the resulting map bears a constant ratio to the area of the sphere. In general, distortion in shape tends to increase toward the poles in latitudes outside of the two standard parallels.

The Albers' projection is well suited to portray areas of large and small east-to-west extent and produces satisfactory results in most cases. However, both standard parallels must lie on the same side of the equator, so this method may not be suitable for map data sets of large north-to-south extent that span the equator. For those map data sets, use the gnomonic projection method.

Figure 23.5 on page 878 illustrates an Albers' equal-area projection of the northern hemisphere.

Figure 23.5 Albers' Projection



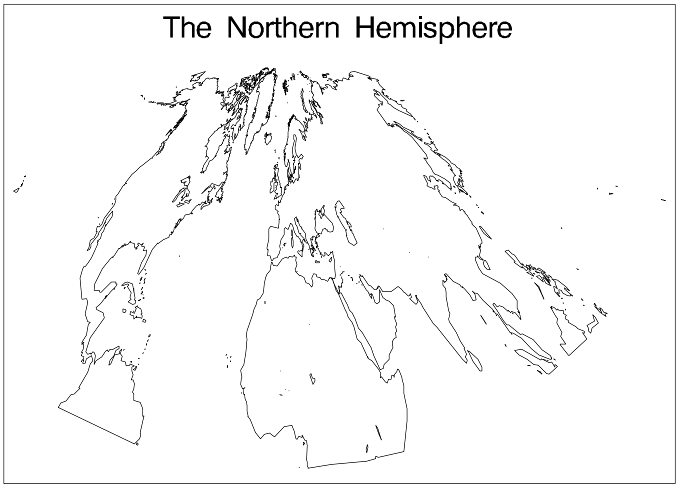
Lambert's Conformal Projection

The *Lambert's projection* is obtained from a secant cone in the same manner as Albers' projection. In the Lambert's projection, meridians of longitude are straight lines that radiate from the apex of the cone, while parallels of latitude are concentric circles. The Lambert's projection is somewhat better than the Albers' projection at representing the original shape of projected unit areas, while the Albers' projection is somewhat better at representing relative sizes of projected unit areas.

The Lambert's projection is ideal for navigational charts and maps of relatively small east-to-west extent. However, as in the Albers' projection, both standard parallels must lie on the same side of the equator, so this method may not be suitable for map data sets that span the equator. For those map data sets, use the gnomonic projection method.

Figure 23.6 on page 878 illustrates a Lambert's conformal projection of the northern hemisphere.

Figure 23.6 Lambert's Projection



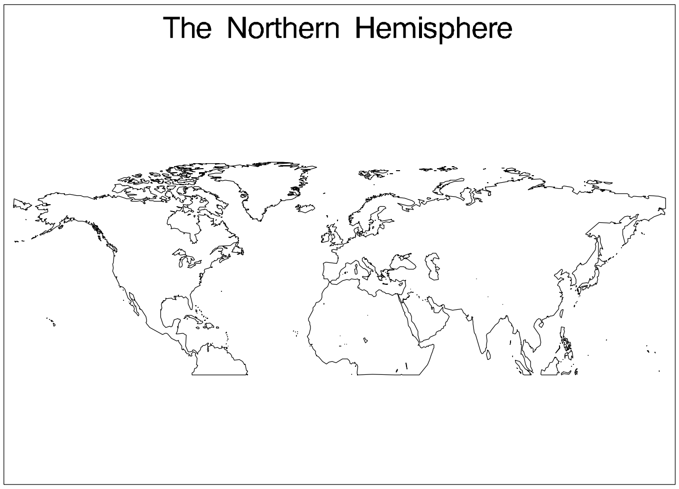
Gnomonic Projection

The *gnomonic projection* is a planar projection from the surface of the sphere directly onto an imaginary plane tangent to the sphere at the map projection pole. By default, the projection pole is placed at the center of the map data set that is to be projected, but you can specify the projection pole to be anywhere on the surface of the sphere. (See POLELONG= and POLELAT= options on page 882.)

In the gnomonic projection, distortion increases as the distance from the map pole increases. Because of this distortion, the PROC GPROJECT procedure deletes all of the observations that lie more than 85 degrees from the map pole. The gnomonic projection is best suited for mapping areas of small east-to-west extent.

Figure 23.7 on page 879 illustrates a gnomonic projection of the northern hemisphere.

Figure 23.7 Gnomonic Projection



Procedure Syntax

Requirements: Exactly one ID statement is required.

PROC GPROJECT <*option(s)*>;
  ID *id-variable(s)*;

PROC GPROJECT Statement

Identifies the input and output map data sets. Optionally specifies the type of projection, and criteria for clipping and projection.

Requirements: An input map data set is required.

Syntax

PROC GPROJECT <*option(s)*>;

option(s) can be one or more options from any or all of the following categories:

- □ data set options:

  DATA=*input-map-data-set*

  OUT=*output-map-data-set*

- □ projection options:

  PARADIV=*n*

  PARALEL1=*latitude*

  PARALEL2=*latitude*

  POLELAT=*latitude*

  POLELONG=*longitude*

  PROJECT=ALBERS | GNOMON | LAMBERT | NONE

- □ coordinate options:

  ASIS | DUPOK

  DEGREE

  EASTLONG

- □ clipping options:

  LATMIN=*min-latitude*

  LATMAX=*max-latitude*

  LONGMIN=*min-longitude*

  LONGMAX=*max-longitude*

Options

ASIS
DUPOK

specify that observations for which the projected values for the X and Y variables are identical to those in the previous observation should be retained. By default, successive identical observations are deleted.

DATA=*input-map-data-set*

identifies the map data set to be processed. By default, the procedure uses the most recently created SAS data set.

See also: “About the Input Map Data Set” on page 875 “SAS Data Sets” on page 25

Featured in: Example 4 on page 891

DEGREE
DEG

specifies that the units for the longitude (X variable) and latitude (Y variable) coordinates are degrees of arc. By default, coordinate units are considered to be radians.

EASTLONG
EAST

specifies that the longitude (X variable) values in the input map data set increase to the east. By default, longitude values increase to the west.

LATMAX=*max-latitude*

specify the maximum latitude that will be included in the projection. Any unit areas that cross the selected latitude are clipped and closed along the specified parallels.

The LATMAX= and LATMIN= options do not have to be paired; you can specify a maximum latitude without specifying a minimum.

When PROJECT=ALBERS, LAMBERT, or GNOMON, PROC GPROJECT treats the value of *max-latitude* as degrees. When PROJECT=NONE, the procedure treats the value as a Cartesian coordinate.

Featured in: Example 3 on page 890

LATMIN=*min-latitude*

specify the minimum latitude that will be included in the projection. Any unit areas that cross the selected latitude are clipped and closed along the specified parallels. The LATMAX= and LATMIN= options do not have to be paired; you can specify a minimum latitude without specifying a maximum.

When PROJECT=ALBERS, LAMBERT, or GNOMON, PROC GPROJECT treats the value of *min-latitude* as degrees. When PROJECT=NONE, the procedure treats the value as a Cartesian coordinate.

Featured in: Example 3 on page 890

LONGMAX=*max-longitude*

specify the maximum longitude to be included in the projection. Any unit areas that cross the selected longitude are clipped and closed along the specified meridians. The LATMAX= and LATMIN= options do not have to be paired; you can specify a maximum longitude without specifying a minimum.

When PROJECT=ALBERS, LAMBERT, or GNOMON, PROC GPROJECT treats the value of *max-longitude* as degrees. When PROJECT=NONE, the procedure treats the value as a Cartesian coordinate.

Featured in: Example 3 on page 890

LONGMIN=*min-longitude*

specify the minimum longitude to be included in the projection. Any unit areas that cross the selected longitude are clipped and closed along the specified meridians. The LATMAX= and LATMIN= options do not have to be paired; you can specify a minimum longitude without specifying a maximum.

When PROJECT=ALBERS, LAMBERT, or GNOMON, the GPROJECT procedure treats the value of *min-longitude* as degrees. When PROJECT=NONE, the procedure treats the value as a Cartesian coordinate.

Featured in: Example 3 on page 890

OUT=*output-map-data-set*

names the new map data set, which contains the coordinates of the new unit areas that are created by the GPROJECT procedure.

By default, the GPROJECT procedure names the new data set that uses 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.

Featured in: Example 4 on page 891

PARADIV=*n*

specifies the divisor that computes the values used for standard parallels for the Albers' or Lambert's projections when explicit values are not provided. By default PARADIV=4, which causes standard parallels to be set at 1/4 and 3/4 of the range of latitude values in the input map data set.

See also: the PARALEL1= and PARALEL2= on page 882 options

PARALEL1=*latitude*

PARALEL2=*latitude*

specify values for the standard parallels that are used in the Albers' or Lambert's projection. *Latitude* must be in degrees. Positive values indicate north of the equator, and negative values indicate south of the equator. These options are ignored for the gnomonic projection.

By default, the GPROJECT procedure calculates values for the standard parallels. The defaults are chosen to minimize the distortion inherent in the projection process. The algorithm used is

PARALEL1 = minlat + R / P_D
PARALEL2 = maxlat - R / P_D
where:

R
is the range of latitude values in the input map data set.

P_D
is the PARADIV= value (see the discussion of the PARADIV= on page 881option).

minlat
is the minimum latitude value in the input map data set.

maxlat
is the maximum latitude value in the input map data set.

If you do not use PARALEL1= or PARALEL2=, or you omit either option, the GPROJECT procedure uses the calculated value for the missing parameter.

The standard parallels, whether explicitly specified or supplied by the procedure, must lie on the same side of the equator. If they do not, PROC GPROJECT prints an error message and stops (the procedure may calculate standard parallels that lie on opposite sides of the equator). When projecting a map data set that contains unit areas that cross the equator, you may have to explicitly specify standard parallels that both lie on the same side of the equatMor. If this causes excessive distortion of the map, you may be able to use the gnomonic projection instead of the Albers' or Lambert's projection because the gnomonic technique has no such limitations at the equator.

POLELAT=*latitude*
POLELONG=*longitude*

specify a projection pole to use for the gnomonic projection. The projection pole is the point at which the surface of the sphere touches the surface of the imaginary plane onto which the map is projected. POLELAT= specifies the latitude of the projection point.

Units for *latitude* are degrees; positive values indicate north of the equator, and negative values indicate south of the equator. POLELONG= gives the longitude for the projection point. Units for *longitude* are degrees; positive values indicate west of the prime meridian, and negative values indicate east of the prime meridian (unless EASTLONG also has been used in the PROC GPROJECT statement).

If you do not use POLELAT= or POLELONG=, or you omit either option, PROC GPROJECT uses values for the position of the center of the unit areas that are defined by the DATA= data set for the missing parameter.

Note: The map that is defined by the input map data set should not contain points more than 85 degrees (1.48353 radians) from the projection pole; all points that exceed this value are deleted from the output map data set. △

Featured in: Example 2 on page 888

PROJECT=ALBERS | LAMBERT | GNOMON | NONE

specifies the projection method to apply to the map data set. Values for PROJECT= are as follows:

ALBERS
specifies Albers' equal-area projection with two standard parallels.

LAMBERT
specifies Lambert's conformal projection with two standard parallels.

GNOMON
specifies the gnomonic projection, which is an azimuthal projection.

NONE
specifies that no projection should be performed. Use this option in conjunction with the LATMIN=, LATMAX=, LONGMIN=, and LONGMAX= options to perform clipping without projection (for example, on map data sets that have already been projected).

By default, PROJECT=ALBERS.

See also: "About Types of Map Projections" on page 876

Featured in: Example 2 on page 888

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 885

Syntax

ID *id-variable(s)*;

Required Arguments

id-variable(s)

specifies one or more variables in the input map data set that identify unit areas. *Id-variable* can be either numeric or character.

Each group of observations with a different ID variable value is evaluated as a separate unit area.

Using the GPROJECT Procedure

The GPROJECT procedure uses a default projection method and default projection criteria to project your map data set. If you do not want to use these defaults, you can use PROC GPROJECT statement options to

□ select the map projection method

□ specify the map projection criteria

□ create a rectangular subset of the input map data set.

The following sections describe how you can use PROC GPROJECT statement options to select your own projection method and projection criteria.

Selecting Projections

Except when projecting map data sets that cover large areas, all three types of projections (Albers', Lambert's, and gnomonic) produce relatively similar results when you use default projection criteria, so you usually do not need to be concerned about which projection method to use when you produce maps solely for graphics output.

However, the default projection criteria may be unsuitable in some circumstances. In particular, the default specifications fail when the map that is being projected extends on both sides of the equator. On other occasions, you may want to select a projection method to achieve a particular effect.

For the Albers' and Lambert's projections, the two standard parallels must both lie on the same side of the equator. PROC GPROJECT stops with an error message if this condition is not met, regardless of whether you explicitly specify parallel values or let the procedure calculate default values. See the descriptions of PARALEL1= and PARALEL2= on page 882 for more information on how to specify the two standard parallels.

Controlling Projection Criteria

For both the Albers' and Lambert's projections, PROC GPROJECT calculates appropriate standard parallels. You can override either or both of these selections if you explicitly specify values for the PARALEL1= or PARALEL2= option. You can influence the selection of default parallels if you use the PARADIV= option. See "Options" on page 880 for more information on these options.

For the gnomonic projection, PROC GPROJECT determines the longitude and latitude of the approximate center of the input map data set area. You can override either or both of these selections if you explicitly specify values for the POLELAT= or POLELONG= option. See "Options" on page 880 for more information.

The clipping options, discussed in "Clipping Map Data Sets" on page 884, can also influence the calculations of the default standard parallels by changing the minimum and maximum coordinate values.

Clipping Map Data Sets

The GPROJECT procedure can create rectangular subsets of the input map data set. This capability provides a way to extract a portion of a larger map if you do not need all the original unit areas for your graph. The procedure enables you to clip unit area boundaries at specified parallels of latitude or meridians of longitude or both. Unit areas that fall completely outside of the specified clipping limits are excluded from the output map data set. Unit areas bisected by the clipping limits are closed along the clipping parallels and meridians, and all points outside of the clipping limits are excluded.

If the input map data set contains the DENSITY variable, any new vertex points and corners that are created by PROC GPROJECT are assigned a DENSITY value of 0 in the output map data set. This enables you to use a subset of the clipped map without using PROC GREDUCE to assign new DENSITY values. (See Chapter 24, "The GREDUCE Procedure," on page 895 for information on how to reduce the number of points that you need to draw a map.)

You can specify the minimum latitude to be retained in the output map data set with LATMIN= and the maximum latitude with LATMAX= . Minimum and maximum longitude values are specified with LONGMIN= and LONGMAX=, respectively. See "Options" on page 880 for more details on these options.

This is how the PROC GPROJECT interprets the clipping longitude and latitude values:

□ If you specify PROJECT=NONE in the PROC GPROJECT statement, the procedure assumes that the input map data set is already projected and the clipping longitude and latitude values are Cartesian coordinates. In this case, LATMAX= and LATMIN= specify the top and bottom edges, respectively, of the area that you want to extract, and LONGMAX= and LONGMIN= specify right and left edges, respectively.

  You must be familiar with the range of values in the X and Y variables in order to select appropriate clipping limits. Use the MEANS or SUMMARY procedure in base SAS to determine the range of values in X and Y. See the *SAS Procedures Guide* for more information.

□ If PROJECT=ALBERS, LAMBERT, or GNOMON, the clipping values are treated as degrees.

Depending on the size and position of the clipped area and the type of projection that is performed, the resulting map may not be exactly rectangular. PROC GPROJECT performs clipping before projection, so the clipped area may be distorted by the projection process.

To produce a clipped area with a rectangular shape, use PROC GPROJECT in two steps:

1. Project the map using the appropriate projection method and projection criteria.
2. Project the map using PROJECT=NONE, and use LATMIN=, LATMAX=, LONGMIN=, and LONGMAX= to clip the map.

See Example 3 on page 890, for an example of clipping an area from an unprojected map data set.

Examples

The following examples illustrate major features of the GPROJECT procedure. Because these examples use map data sets that are supplied with SAS/GRAPH , you may need to replace *SAS-data-library* in the LIBNAME statement with the actual location of the SAS data library that contains the Institute-supplied map data sets on your system. Contact your SAS Software Consultant for the location of the map data sets at your site. If your site automatically assigns the libref MAPS to the SAS data library that contains the Institute-supplied map data sets, delete the LIBNAME statement in these examples.

Example 1: Using Default Projection Specifications

Procedure features:
ID statement

Other features:
LIBNAME statement

Sample library member: GR23N01

This example demonstrates the effect of using PROC GPROJECT on an unprojected map data set without specifying any options. Because PROJECT= is not used in the PROC GPROJECT statement, the Albers' equal-area projection method is used by default. PROC GPROJECT supplies defaults for the standard parallels that minimize distortion of the projected map areas.

Figure 23.8 Map before Projection (GR23N01(a))

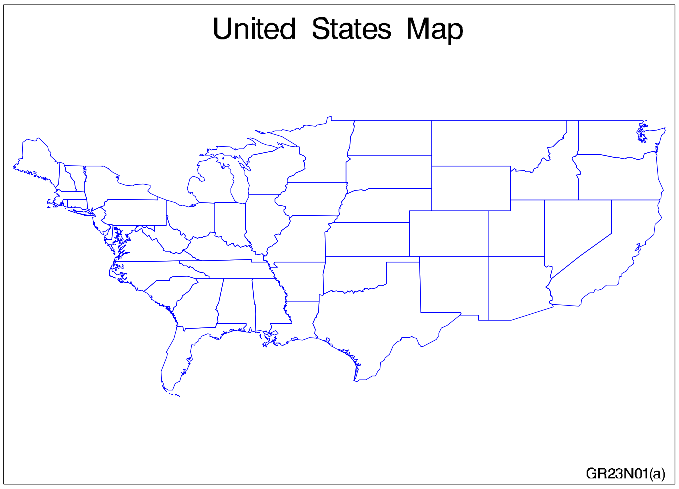


Figure 23.8 on page 886 illustrates the output produced the REFLIB.US48 map data set, which contains only unprojected values, X and Y. Output 23.1 on page 886 shows the variables in the data set.

Output 23.1 The REFLIB.US48 Data Set

```
                         US48 Data Set

        OBS     STATE     SEGMENT     DENSITY        X          Y

          1       1          1           3       1.48221    0.56286
          2       1          1           3       1.48226    0.56234
          3       1          1           3       1.48304    0.56231
     .
     .
     .
```

The GPROJECT procedure is used with the REFLIB.US48 map data set as input to create the projected map data set, REFLIB.US48PROJ. The values for X and Y in this new data set are projected (cartesian). Output 23.2 on page 887 shows the variables in the data set.

Output 23.2 The REFLIB.US48PROJ Data Set

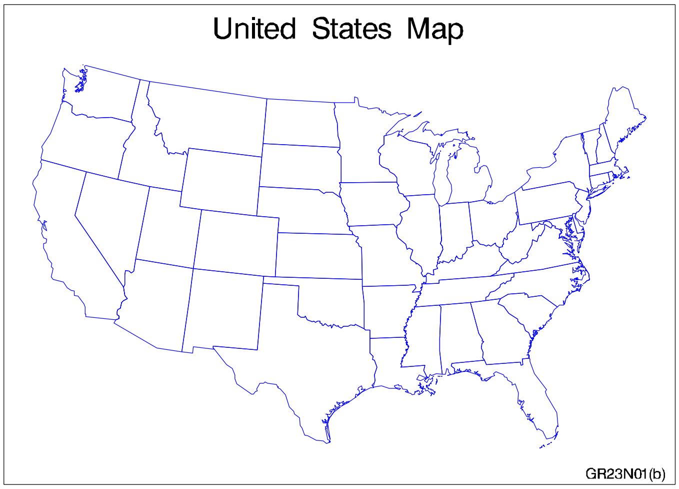
```
                          US48PROJ Data Set

         OBS        X            Y          DENSITY     STATE     SEGMENT

           1     0.16068     -0.073470         3          1          1
           2     0.16069     -0.073993         3          1          1
           3     0.16004     -0.074097         3          1          1
           .
           .
           .
```

The new projected map data set, REFLIB.US48PROJ, is used to create the projected map, Figure 23.9 on page 887.

Figure 23.9 Map after Projection (GR23N01(b))



Assign the librefs and set the graphics environment.

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

Create reduced continental U.S. map data set and remove Alaska, Hawaii, and Puerto Rico.

```
data reflib.us48;
   set maps.states;
   if state ne 2 and state ne 15 and state ne 72;
   if density<4;
run;
```

Define title and footnote for unprojected map.

```
title ’United States Map’;
footnote j=r ’GR23N01(a) ’;
```

Define pattern characteristics.

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

Show unprojected map.

```
proc gmap map=reflib.us48 data=reflib.us48 all;
   id state;
   choro state / nolegend;
run;
```

Project map data set using all default criteria. The ID statement identifies the variable in the input map data set that defines unit areas.

```
proc gproject data=reflib.us48
              out=reflib.us48proj;
   id state;
run;
```

Define footnote for projected map.

```
footnote j=r ’GR23N01(b) ’;
```

Show projected map.

```
proc gmap map=reflib.us48proj
          data=reflib.us48proj all;
   id state;
   choro state / nolegend;
run;
quit;
```

Example 2: Emphasizing Map Areas

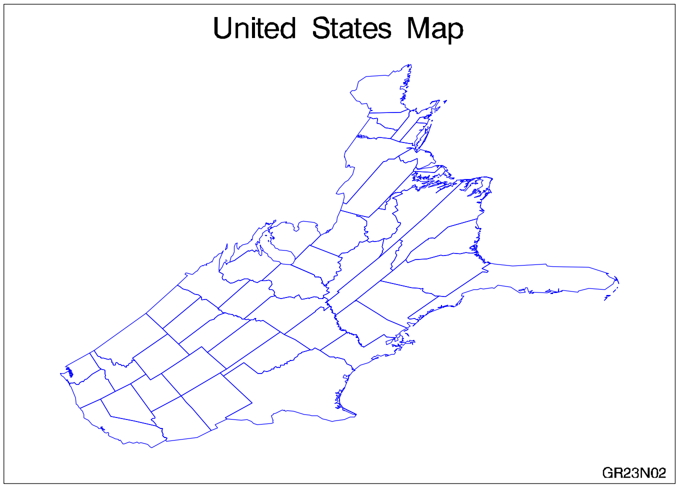
Procedure features:

PROC GPROJECT options:

POLELAT=
POLELONG=
PROJECT=

Data set: REFLIB.US48 on page 887

Sample library member: GR23N02



This example uses the gnomonic projection method to create a map in which the east coast of the United States appears disproportionately large compared to the west coast.

Assign the librefs and set the graphics environment.

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

Project map onto a plane centered in the Pacific. PROJECT= specifies the projection method for the map data set. POLELONG= and POLELAT= specify a projection pole for the gnomonic projection. In this example, the pole is positioned in the Pacific Ocean.

```
proc gproject data=reflib.us48
              out=reflib.skew
              project=gnomon
              polelong=160
              polelat=45;
   id state;
run;
```

Define title and footnote for the map.

```
title 'United States Map';
footnote j=r 'GR23N02 ';
```

Define pattern characteristics.

```
pattern value=mempty repeat=49 color=blue;
```

Show the projected map.

```
proc gmap map=reflib.skew data=reflib.skew all;
   id state;
   choro state / nolegend;
run;
quit;
```

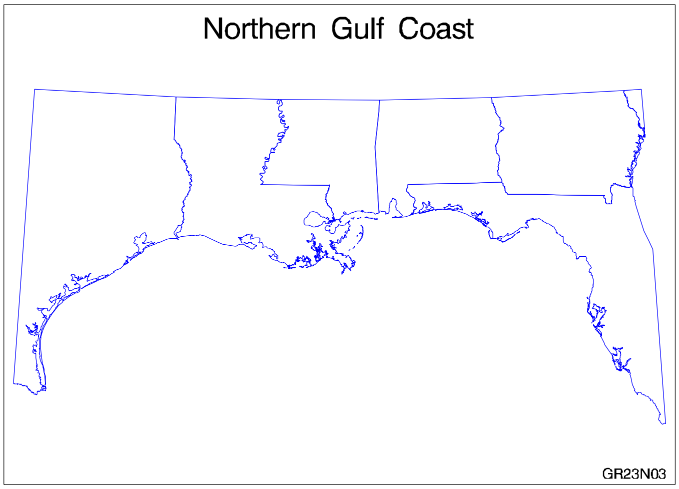
Example 3: Clipping an Area from the Map

Procedure features:

PROC GPROJECT options:

LONGMAX=
LONGMIN=
LATMAX=
LATMIN=

Sample library member: GR23N03



This example uses the clipping capabilities of PROC GPROJECT to create a map of the states in the United States that border the Gulf of Mexico. Because the PROJECT= option is not used in the GPROJECT procedure, the Albers' equal-area projection method is used by default.

Assign the librefs and set the graphics environment.

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

Clip and project rectangular subset of the map. LONGMIN= and LONGMAX= specify the minimum and maximum longitudes to be included in the map projection. LATMIN= and LATMAX= specify the minimum and maximum latitudes to be included in the map projection.

```
proc gproject data=maps.states
              out=reflib.gulf
              longmin=81
              longmax=98
              latmin=25
              latmax=33;
   where density<5;
   id state;
run;
```

Define title and footnote for the map.

```
title 'Northern Gulf Coast';
footnote j=r 'GR23N03 ';
```

Define pattern characteristics.

```
pattern value=mempty repeat=7 color=blue;
```

Show the clipped map.

```
proc gmap map=reflib.gulf data=reflib.gulf all;
   id state;
   choro state / nolegend;
run;
quit;
```

Example 4: Projecting an Annotate Data Set

Procedure features:

PROC GPROJECT options:

DATA=
OUT=

ID statement

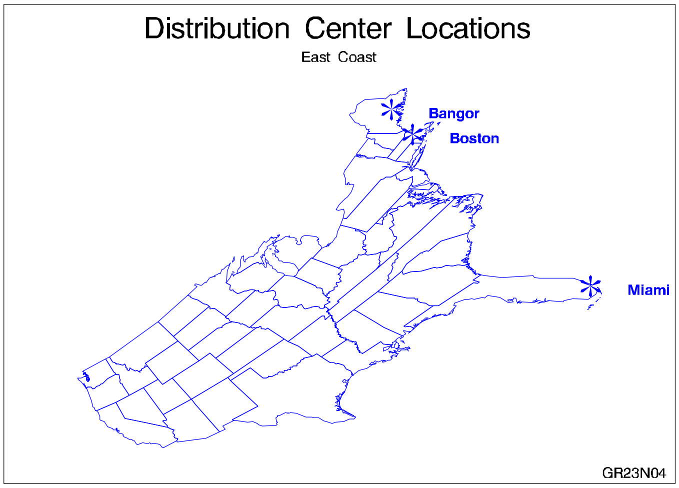
Other features:

CHORO statement

Annotate data set

Data set: REFLIB.US48 on page 887

Sample library member: GR23N04



This example illustrates how to project an Annotate data set for use with a map data set. It labels the locations of Miami, Boston, and Bangor on the map shown in the second example. Because the X and Y variables in the USCITY data set already have been projected to match the US data set, they cannot be used with the map that is produced by the second example. To properly label the projected map, the example uses the same projection method for the city coordinates as the method that is used for the map coordinates. This example illustrates how to use the same projection method for both data sets.

Assign the librefs and set the graphics environment.

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

Create Annotate data set CITIES from the MAPS.USCITY data set. The unprojected LONG and LAT variable values are converted to radians and substituted for the projected X and Y variable values. LONG and LAT are converted by multiplying them by the arccosine of –1 and dividing that amount by 180. The cities are each assigned a value for the NEWST variable, sequentially beginning at 100.

```
data cities(drop=state rename=(newst=state));
   set maps.uscity(keep=lat long city state);
   length function style color $ 8
          position $ 1 text $ 20;
   retain function 'label' xsys ysys '2'
```

```
         hsys '1' when 'b' newst 100;
   if state=12 and city='Miami' or
      state=25 and city='Boston' or
      state=23 and city='Bangor';
   newst+1; color='blue'; size=10; text='T';
      position='5';
      style='marker'; x=long*arcos(-1)/180;
      y=lat*arcos(-1)/180; output;
   newst+1; color='blue'; size=4;
      text='     '||city;
      position='6'; style='swissb'; output;
run;
```

Create data set ALL by combining data set REFLIB.US48 and data set REFLIB.CITIES.

```
data all;
   set reflib.us48 cities;
run;
```

Project the ALL data set. DATA= specifies the data set to be projected. OUT= specifies the name of the new projected data set that is created. The ID statement identifies the variable in the input map data set that defines map areas.

```
proc gproject data=all
              out=allp
              project=gnomon
              polelong=160
              polelat=45;
   id state;
run;
```

Separate the projected data set into the CITIESP Annotate data set and the US48P map data set.

```
data citiesp us48p;
   set allp;
   if state>100 then output citiesp;
   else output us48p;
run;
```

Define title and footnote for the map.

```
title1 'Distribution Center Locations';
title2 'East Coast';
footnote j=r 'GR23N04 ';
```

Define pattern characteristics.

```
pattern value=mempty repeat=49 color=blue;
```

Show the annotated map. The CHORO statement displays the projected map and annotates it using the projected Annotate data set.

```
proc gmap data=us48p map=us48p all;
   id state;
   choro state
         / nolegend
           annotate=citiesp;
run;
quit;
```

References

Pearson, F., II (1977), "Map Projection Equations," Report Number TR-3624, Naval Surface Weapons Center, Dahlgren Laboratory, March, 1977.

Richardus, P. and Adler, R.K. (1972), *Map Projections*, Amsterdam: North-Holland Publishing Company; New York: American Elsevier Publishing Company.

Robinson, A.H. (1978), *Elements of Cartography*, New York: John Wiley & Sons, Inc.

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

ISBN 1-58025-525-6

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

1st printing, October 1999

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