

Generating Thematic Maps with Proportional Shading and Graphical Annotation

Larry Hoyle, University of Kansas Institute for Public Policy and Business Research

Overview

SAS[®] provides a number of drivers for devices capable of nearly continuous shading. Some of these devices are commonly available. These include PostScript printers and VGA displays. PostScript output can also be previewed on UNIX workstation displays and sent to high-quality typesetters. This paper describes one application which takes advantage of these devices' capability - generation of choroplethic maps with shading proportional to some value.

The example used in this paper will show the mapping of population figures for the states in the MWSUG region. This map was prepared using SAS 6.04 under MSDOS on a 386 system with VGA display. The printed map was previewed on an Apple Laserwriter and final copy was produced on a Linotronic 300.

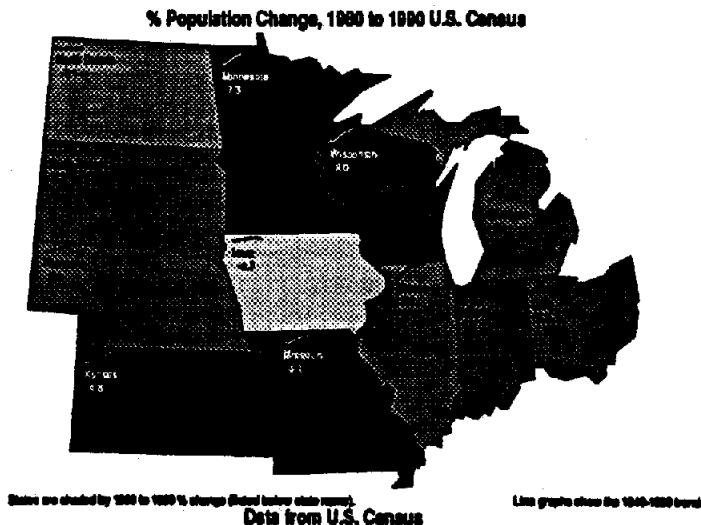


Figure 1

If the map is to have state names or values listed you must first make an annotation file. Since this file may be used for a number of different maps it may be efficient to make it a permanent SAS dataset. Making an annotation file for a new collection of regions (eg. counties or states) will probably take a few iterations to get names positioned just right. A simple technique gives a good first try, and then labels can be adjusted individually.

Once the annotation file is built, the mapping of a variable can proceed. The first step in the mapping is a (mathematical) mapping. SAS provides several color spaces which can be used for proportional shading. The RGB, HLS and GRAY color spaces all provide enough levels of shading to shade proportionally. For the devices used in this example the GRAY scale was appropriate since none

are color devices. If the variable to be mapped is highly skewed, it may be transformed before using it to shade, but in many cases a linear mapping is acceptable.

Table 1 shows some sample values mapped first into a set of integers

Value	Ival	Proportional Shade	Sequential Shade
-3.91	32	grayF1	grayFF
0.41	108	grayA5	grayEE
0.94	118	gray9B	grayDD
1.30	124	gray95	grayCC
1.91	135	gray8A	grayBB
2.00	136	gray89	grayAA
2.39	143	gray82	gray99
4.12	174	gray63	gray88
5.76	202	gray47	gray77
5.75	202	gray47	gray66
7.20	228	gray2D	gray55
7.95	241	gray20	gray44

Table 1

which are then mapped into proportional shade values. The last column shows shade values which might be assigned using a fixed color list. In the latter case each state still gets a unique shade but the difference in shading between the first and second state is the same as the difference between the second and third state.

Notice that in the proportional shading column the lowest value maps into the highest gray value used, grayF1. GrayFF is supposed to be white and gray00 should be black, but with some drivers the colors near the ends may not be what you'd expect. The lowest gray value used here is gray20 which is almost black. If absolute black is used, the state borders can disappear between adjacent dark regions.

Kansas County Average Precipitation (Inches), 1951-80

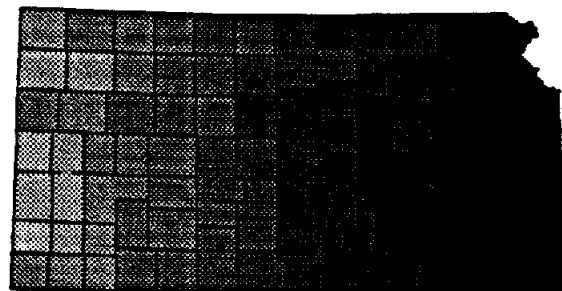


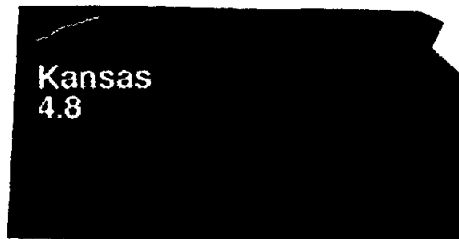
Figure 2

Generating proportional shading involves writing out the shading values to be included into a COLOR statement later in the SAS program. This color statement is then cycled through for one PATTERN statement to shade the regions.

At this point a simple shaded map can be generated with proc GMAP (see Figure 2.) This map may be annotated with names and values. One might also use

the annotation facility to display the distribution of the variable.

If the shading variable is part of a time series, then it may also be informative to display a trend line within each region. This line can be added to the annotation without too much effort. Figure 1 shows trend lines for population from 1940 to 1990 above each state name. Figure 3 shows a detail from that map.



Line graphs show the 1940-1990 trend.
Figure 3

Building the Basic Annotation File

Listing 1 shows the SAS program which generates the annotation file which places state names within each state. A data step produces the dataset MWSTATES which contains the state FIPS code and the mixed case name of the state. A PROC SUMMARY then creates a dataset MWMINMAX with the minimum and maximum values for the x and y variables in the SAS supplied state boundary file for the states to be mapped. These two files are then merged into the permanent annotation file for this collection of states. The standard annotation variables are added during this last data step, as described in the *SAS/Graph Software manual*.

OBS	4
STATE	20
MINX	-0.08591
MINY	-0.00403
MAXX	0.01698
MAXY	0.05101
POSITION	9
FUNCTION	LABEL
STYLE	NONE
WHEN	A
X	-0.07871
Y	0.04441
XSYS	2
YSYS	2
HSYS	3
TEXT	Kansas
ANNOTYPE	NAME
SIZE	1.75
LINE	1

Table 2 One Record From
MAPS.MWANNO

Listing 1

```

/*-----*/
/* MKANO makes the annotation file used by the mapping program. */
/* It reads a list of states to be included, along with their */
/* mixed case names. A separate annotation file is needed for */
/* each region to be mapped. */
/* Larry Hoyle, University of Kansas, IPPBR */
/*-----*/

/*-----*/
/* First read in the list of states in the region. */
/*-----*/

DATA MWSTATES;
LENGTH STATE 3;
INPUT STATE $NAME $20.;
CARDS;
17 Illinois
18 Indiana
19 Iowa
20 Kansas
26 Michigan
27 Minnesota
29 Missouri
31 Nebraska
38 North Dakota
39 Ohio
46 South Dakota
55 Wisconsin
;

/*-----*/
/* Read the state boundaries SAS provides (MWSUG STATES ONLY). */
/* Compute the extrema for the states. */
/*-----*/

PROC SUMMARY NWAY DATA=MAPS.US;
WHERE STATE IN(17,18,19,20,26,27,29,31,38,39,46,55);
CLASS STATE;
VAR X Y;
OUTPUT OUT=MWMINMAX MIN=MINX MINY MAX=MAXX MAXY;
/*-----*/
/* Add the variables which make an annotation file */
/*-----*/

DATA MAPS.MWANNO;
MERGE MWSTATES MWMINMAX;
BY STATE;
DROP $NAME;
LENGTH POSITION $ 1;
LENGTH FUNCTION $ 8;
LENGTH STYLE $ 8;
LENGTH WHEN $ 1;
LENGTH X $ 8;
LENGTH Y $ 8;
LENGTH XSYS $ 1;
LENGTH YSYS $ 1;
LENGTH HSYS $ 1;
LENGTH TEXT $ 12;
FUNCTION=LABEL;
/* PUT STATE NAMES BELOW AND LEFT ALIGNED */
ANNOTYPE=NAME;
POSITION=9;
STYLE=NONE;
TEXT=$NAME;
WHEN=A;
/* Annotate after shading. */
XSYS=2;
/* X and y coordinate system is */
YSYS=2;
/* within the data area. */
HSYS=3;
/* SIZE is percent of total screen. */
SIZE=1.75;
/* thin line */
LINE=1;
DROP _TYPE_ _FREQ_;
```

The x and y variables are initially adjusted relative to the minimum and maximum values for x and y in the SAS boundary file. To put the names in the upper left corner of each state add about 7% of the state width (maximum of x minus minimum of y) to x and subtract 7% of the height from y. After displaying the map for the first time, additional adjustments will be needed. If you continue to adjust as a proportion of the state height or width then scaling will not change the relative placement within the state. For states too small for a label, the label can be placed outside the state and a line between the state center and the label can be added to the annotation file. An example of code which will generate such a line is included as a comment at the end of Listing 1.

Shading the Map

The macro SHADES generates the color list which is the key to being able to shade proportionally. Other color schemes than grayXX can be used by changing the "shade =" statement in the macro. For a color device you could use a continuum ranging from yellow-brown at one end to bright green at the other to represent something like rainfall. This could be done with an appropriate equation using the RGB color space. The same "shade=" statement could be rewritten to switch the direction of gray shading from highest value is darkest to highest value is lightest.

The map program displayed in Listing 2 begins with two macro variables. The first, ANNOFMT, sets the format for values displayed in the states. The second, SCALFAC, is used to scale the size of the graph within each county. A filename statement selects the file to which the PostScript will be written. Title and footnote statements follow the filename statement.

```
X=MINX + (MAXX-MINX)*.07; /* 7% EAST OF WEST EDGE */
Y=MAXY - (MAXY-MINY)*.07; /* 7% SOUTH OF NORTH EDGE */
/* MAKE ADJUSTMENTS FOR ODD STATES */
```

```
F state = 17 THEN X = X + (MAXX-MINX)*.30; /* ILLINOIS */
F state = 18 THEN Y = Y - (MAXY-MINY)*.08; /* INDIANA */
F state = 26 THEN X = X + (MAXX-MINX)*.50; /* MICHIGAN */
F state = 26 THEN Y = Y - (MAXY-MINY)*.50;
F state = 20 THEN Y = Y - (MAXY-MINY)*.05; /* KANSAS */
F state = 27 THEN Y = Y - (MAXY-MINY)*.07; /* MINNESOTA */
F state = 29 THEN X = X + (MAXX-MINX)*.15; /* MISSOURI */
F state = 39 THEN Y = Y - (MAXY-MINY)*.20; /* OHIO */
F state = 55 THEN X = X + (MAXX-MINX)*.05; /* WISCONSIN */
F state = 55 THEN Y = Y - (MAXY-MINY)*.10;
```

```
/* WRITE OUT THE STATE LABEL */
```

```
OUTPUT;
```

```
/* THEN DRAW ANY LINES TO EXTERNAL LABELS */
```

```
/* ----- */
/* IF state = ?? THEN DO; */
/* SIZE=1; LINE=2; */
/* FUNCTION=MOVE; X=(MAXX+MINX)/2-(MAXX-MINX)*.25; */
/* Y=(MAXY+MINY)/2+(MAXY-MINY)*.20; */
/* OUTPUT; */
/* FUNCTION=DRAW; X=MAXX-(MAXX-MINX)*.15; */
/* Y=MAXY+(MAXY-MINY)*.15; */
/* HSY=4; OUTPUT; */
/* END; */
/* ----- */
```

```
PROC PRINT;
```

```
RUN;
```

Listing 2

```
/* ----- */
/* P809MW.SAS makes a proportionally shaded map of the MWSUG states. */
/* Data are from the U.S. Census. States are shaded in proportion to */
/* population change from 1980 to 1990. The annotation includes a */
/* graph of change over 5 decades for each state. */
/* The program creates these temporary datasets: */
/* RAWPLUS - the user dataset with shading variables added. */
/* PREANNO - the annotation file with values, names, and graphs */
/* MWSUG - the state boundaries for the MWSUG region */
/* The output PostScript file is named with the filename GRAFOUT */
/* Larry Hoyle, University of Kansas, */
/* Institute for Public Policy and Business Research */
/* ----- */

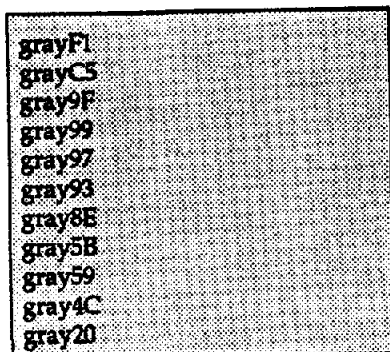
%global annofmt scalfac;
/* %global locolor hicolor; */

%let annofmt=9.1;
%let scalfac=0.0014; /* SCALFAC scales the size of the line graph */
/* it may be set relative to the height of */
/* average state. */

filename grafout 'mwsug.ps';

title1 h=9 pct ' ';
title2 F=none C=black H=3
'% Population Change, 1980 to 1990 U.S. Census';
footnote1 h=3 c=black f=none
'States are shaded by 1980 to 1990 % change (listed below state name).';
'Line graphs show the 1940-1990 trend.';
footnote2 f=none c=black h=3
'Data from U.S. Census';
footnote3 h=10 pct c=black ' '; /* required to place map on page */
```

SHADES can take 5 arguments. The variable used for shading and the dataset containing it are necessary. A separate variable can be specified as a label variable. This is useful in cases where shading is proportional to a transformation of the variable of interest. The final two macro variables specify the lowest and highest (decimal) values to use for a color. These can be used to adjust for some of the drivers with odd sequences of shades at the low and high ends, or to adjust overall lightness of the map.



grayF1
grayC5
gray9F
gray99
gray97
gray93
gray8E
gray5B
gray59
gray4C
gray20

Table 3 The Contents of File Shades

The macro creates a dataset named RAWPLUS which is the input dataset with 4 variables added. The new variable RVAL is a copy of the variable the user selects to use for shading. The variable IVAL contains the integer value which corresponds to the shade for the real value. IVAL is used by PROC GMAP as the discrete grouping variable. The variable LABLVAR is used in a following data step to annotate the map with values of the shading variable. If a variable name is not specified for LABLVAR when the macro is invoked, LABLVAR is given the name of the shading variable. The variable SHADE contains a string which specifies the grayXX value for the state. Adding these variables to the dataset allows the SAS code which follows the macro to be independent of the names of the actual input dataset and shading variable.

The macro also writes a file named SHADES which contains the unique grayXX values for the color statement which follows invocation of the macro. The list of values is included back into the stream of SAS code when needed.

Following the macro definition in Listing 2 is a data step which reads the values to be mapped. These values are U.S. Census Total Population from 1940 to 1990. Percent changes from 1940 to each other year and from 1980 to 1990 are computed. Invocation of the macro SHADES follows this data step.

Graphics options follow the invocation of the macro SHADES. The most crucial of these options is the COLORS= statement into which the color list from SHADES is %included. This list defines the sequence of colors PROC GMAP will use when shading the map. Since

```

/* generate a continuous shaded map */
/* ----- */
/* macro SHADES
/* This macro is used for generating "continuous shaded" maps. It maps
/* a variable into integers in the right range for gray colors for
/* a PostScript printer. It also generates a list of color names.
/* Inputs are:
/* var= the variable to use for shading.
/* lablvar= the variable to print in each state.
/* data= the data set containing these variables
/* loshade= the lowest shade value to use
/* hishade= the highest shade value to use
/* Output is a dataset named RAWPLUS with the original variables plus:
/* rval the value of the shading variable (real)
/* lablvar the value to be used for labeling
/* ival rval mapped onto the interval loshade->hishade
/* shade the "GRAYXX" color value to be used
/* and an output file named SHADES with values for a COLOR statement */
/* ----- */

%macro shades(var=, lablvar=, data=, loshade=32, hishade=241);
    /*let locolor=&loshade; */
    /*let hicolor=&hishade; */

    %if %length(&lablvar)=0 %then %do;
        %let lablvar = &var;
    %end; /* %if %length(&lablvar) */

    proc sort data=&data; by &var;
    proc summary data=&data;
        var &var;
        output out=minmax min(&var)=xmin max(&var)=xmax;
    proc print data=minmax;
    run;

    data rawplus;
        set &data;
        if _n_ = 1 then set minmax;
        drop _type_; /* ----- */
        /* RVAL contains the real value of the plot */
        rval=&var; /* variable to be used for sorting. */
        /* LABLVAR contains a value for labeling. */
        lablval=&lablvar; /* IVAL contains the integer transformed value. */
        /* ----- */
        ival=int((.499999 + &loshade +
            (&var-xmin) * ((&hishade-&loshade)/(xmax-xmin)))
            /*reverse the order */
            shade = 'gray' || put(&hishade - ival + &loshade, hex2.);

    data _null_;
        set rawplus;
        by ival;
        file shades;
        if first.ival then put shade;

    run;
%mend shades;
/* ----- */
/* End the macro shades.
/* ----- */

/* ----- */
/* ENTER THE DATASET TO BE MAPPED HERE.
/* Note: the user dataset should not have variables named:
/* RVAL, LABLVAR, IVAL, or SHADE.
/* ----- */

data MWSPOP;
length state 3;
input post $ state p40 p50 p60 p70 p80 p90;
/* ----- */
/* Variables pNN50 are percent change from 1950 to 19NN.
/* ----- */
p5040 = 100 * (p50-p40) / p40;

```

there is one color for each unique value of IVAL, this list will assign the appropriate color to each state.

obs	function	x	y	size
25	LABEL	-0.078705	0.043009	1.75
26	LABEL	-0.078705	0.036009	1.75
27	MOVE	-0.078705	0.043009	1.75
28	draw	-0.075905	0.043817	2.00
29	draw	-0.073105	0.045947	2.00
30	draw	-0.070305	0.046491	2.00
31	draw	-0.067505	0.047385	2.00
32	draw	-0.064705	0.048271	2.00

Table 4 Selected Variables from Dataset PREANNO for Kansas

Annotating with Names, Values, and Graphs

The MAPS.MWANNO annotation file which was built earlier serves as a base for the final annotation file used for the map. A new annotation file, PREANNO, is created by merging MAPS.MWANNO with the dataset RAWPLUS. State names are already placed in this file but may be black on black shading. Using the value from RAWPLUS, the color of the state name is inverted if necessary and then output to the new dataset.

The y variable is then decremented by the value of &SCALFAC to place the a value for the state on the map. This record is then output to PREANNO.

To generate line graphs, the dataset first computes a set of variables y50 through y90 which contain the vertical position for the line at each year after 1940. Variable y is then set to the vertical position for the beginning of the line and a FUNCTION='MOVE' record is output to PREANNO.

A sequence of FUNCTION='DRAW' records is then written in a loop which ranges over an array of the vertical position variables. Table 4 shows values from PREANNO for the state of Kansas. Observation 25 is the state name and has a TEXT value of 'Kansas'. Observation 26 is the value and has a TEXT value of '4.8'. Observations 27 through 32 draw the graph above the state name.

The first PATTERN statement shades the map. The second is never used unless there are missing data. A PROC GMAP draws the map. The DISCRETE option allows each unique value of IVAL to have its own shade.

```
p6040 = 100 * (p60-p40) / p40;
p7040 = 100 * (p70-p40) / p40;
p8040 = 100 * (p80-p40) / p40;
p9040 = 100 * (p90-p40) / p40;
```

```
p9080 = 100 * (p90-p80) / p80;
cards;
      / population (thousands) */
11 17 7897 6712 10081 11110 11427 11430
18 18 3428 3634 4882 5195 5490 5544
19 19 2538 2621 2758 2825 2914 2777
20 20 1801 1905 2179 2249 2384 2478
26 26 5258 6372 7823 8882 9262 9295
27 27 2792 2982 3414 3806 4076 4375
29 29 3785 3955 4320 4678 4917 5117
31 31 1316 1326 1411 1485 1570 1578
38 38 642 620 632 618 653 639
39 39 6908 7947 9708 10857 10798 10847
46 46 643 653 681 666 691 698
55 55 3138 3436 3952 4418 4706 4892
;
```

```
/*
/* Invoke the macro.
/*
```

```
%shades(data=MAPSPOP, var=p9080)
```

```
options gsfname=gratout gsfmode=replace;
options device=aplus nodisplay gellen=132 penmounts=255;
options rotate;
options characters chartype=11; /* 9 is helvetica, 11 helv bold */
      /* note SAS scales Helvetica as wide characters */
      /* use hpos=200 in landscape to narrow them */
options hpos=200;
options colors=() /* pull in the color values written by the macro */
%include 'shades';
);
```

```
proc sort data=maps.mwanno; by state;
proc sort data=rawplus; by state;
```

```
/*
/* Build the annotation file.
/*
```

```
data preanno;
length color $8;
merge maps.mwanno rawplus
(keep=state ival rval lablval shade
p50 p60 p70 p80 p90 p5040 p6040 p7040 p8040 p9040 p9080);
BY STATE;
/*
/* Put black names on light shading, white names on dark.
/*
size=1.75;
y = y - (&scalfac);
color='black';
if ival > 140 then color = 'white';
output;
```

```
/*
/* Place the values below the name.
/*
```

```
if upcase(function) = 'LABEL' then do;
text = ' ' || left(pu(lablval, &annotmt));
y = y - (&scalfac)*5;
output;
/*
/* Line graphs within counties, above the name.
/*
y = y + ((&scalfac)*5); /*
```

Reproducing Shading

Once you have a map, getting to the final print stage may take some thought. With a PostScript output file, for example, you may have the option of producing camera ready copy on a laser printer with 300 dot per inch (dpi) resolution or on a typesetter with 1200 or 2400 dpi resolution. A 300 dpi printer will actually generate only about 50 different levels of shading. Names and values printed over intermediate shades will appear fuzzy when the halftoning dots fall on the edges of the letters. Resolutions of 1200 or 2400 will produce crisp text at small sizes over gray tones and will produce many more levels of shading.

If the original is to be photocopied, though, 300 dpi may give the best results. If the map is to be offset printed, photographing a paper camera-ready copy may not reproduce the differences in shading adequately. This may be a particular problem if the photographic process changes the size of the image.

The best plan, if possible, is to produce a negative directly from the typesetting equipment. Listing 3 shows a segment of PostScript code which can be placed at the beginning of a single page PostScript file which will convert the image to a negative.

Your layout or word processing software may also be capable of printing negatives. Using The SAS encapsulated PostScript drivers you can produce a map which can be pulled into your word processor and scaled to whatever size you want.

Conclusions Maps and More

SAS provides the tools for proportionally shaded maps of U.S. states and counties, Canadian provinces, and countries of the world. With boundary files from other sources such as the U.S. Census Tiger files, shaded maps can be generated for a wide variety of other geographic units.

The technique of writing out a color list and reincluding it might also work with other SAS procedures which use the PATTERN statement. Other SAS procedures may support color values directly. The SCATTER statement within PROC3D, for example allows for specification of a variable containing a color value. This value could be directly proportional to some variable of interest.

Trademark Notice

SAS and SAS/GRAPH is the registered trademark of SAS Institute, Inc. Cary, NC, USA. PostScript is a trademark of Adobe Systems Incorporated.

References

Roth, Stephen F. *Real World PostScript*, Addison-Wesley Publishing Company, 1988.

Author

Larry Hoyle
Institute for Public Policy and Business Research
University of Kansas
607 Blake Hall
Lawrence, Kansas 66045-2960
Telephone: (913) 864-3701
Internet: LHOYLE@UKANVM.CC.UKANS.EDU

```

y50 = y + (p5040/10) * &scalac; /* 30 % is about max change */
y60 = y + (p6040/10) * &scalac; /* making maximum height */
y70 = y + (p7040/10) * &scalac; /* 3 times scalac */
y80 = y + (p8040/10) * &scalac; /* ----- */
y90 = y + (p9040/10) * &scalac;

function = 'MOVE';
output;

function = 'draw';
size=2;

array yy[5] y50 y60 y70 y80 y90;
do ix=1 to 5;
  x=x+(&scalac)*2;
  y=yy[ix];
  output;
end;

end; /* uppercase(function) = 'label' */
/* ----- */
/*      Select the MWSUG states from the boundary file. */
/* ----- */

data MWSUG;
set maps.us;
where state in(17,18,19,20,26,27,29,31,36,39,46,55);

/* ----- */
/*      Finally, draw the map. */
/* ----- */

PATTERN1 V=s;
PATTERN2 v=m1x;
PROC GMAP DATA = rawplus
  MAP = mwsug
  ANNO = preanno
  ALL;
ID state;
CHORO hval / DISCRETE
  MISSING
  NOLEGEND
  CTEXT = black
  CEMPTY = white
  COUTLINE = black;

RUN;

```

Listing 3

```

% convert everything to negative
/initial-xfer currenttransfer def
{ 1 exch sub initial-xfer bind settransfer
% black out the background from Real World Postscript p. 70
gsave
newpath
clippath %current clipping path should be the page
1 setgray %this is now black
% use stroke for testing
% stroke
fill
grestore
% so that the emulsion is on the right side,
% flip the image (this only works for an 8.5 inch page)
% -1 0 0 1 multiply x values by -1
% 812 0 add 812 points to the values (8.5 inches)
[-1 0 0 1 812 0] concat
% this ends convert to negative section

```