

Beyond the Standard Map

Alternate Methods for Visualizing Location Data

Seth Hoffman, GEICO

ABSTRACT

The bulk of a data scientist's job is data preparation and cleaning. However, presenting the findings in a compelling manner is what generates speaking engagements. "A Map!" is the obvious answer for visualizing location data. Though the definition is quite broad, most people picture a general reference map, or maybe a heat map if it is the middle of summer. This paper uses Proc GMAP as the drawing tool to explore some less traditional metaphors and ideas for displaying analysis results which contain a spatial component.

INTRODUCTION

Effectively telling a story with an images requires knowing what information truly needs to be understood by the viewers. Line graphs, bar charts, swim-lane diagrams, and a wide array of other options are available for adding to displays. Each of these types of graphics convey information through their different properties. Lengths and widths, colors, and area are the primary variables at play in all of these. However, with geographic displays, direction, contiguity, and actual location become tools to add to the storytelling toolbox.

EVERYBODY DOES THIS

When thinking about presenting location information, two types of visualizations invariably come to mind: an aerial image or a standard map.

Everybody can look at a "real color" satellite image and understand what it is they are seeing. Each pixel of the image is the average value of all the light coming from some patch of ground. Depending on your security clearance level, that area can vary from sub-square meter to a square kilometer or more. If the picture is of a region known to the viewer, they may be able to apply some geographic context to the presented information.

Satellite imagery can be acquired from any number of websites¹. Once saved as an image file, example code ² shows how easy it is to use as a background canvas on which to put additional information. Even though the picture might look nice, the question always has to be: "does it help communicate the story?" Unless the data is related to the type of land cover, it might be more of a distraction than an aide. The story being told is dependent not just on the data, but also on the viewer's knowledge of the locations of those points.



Figure 1a and 1b. Generic image + generic data = no added value

Figure 1a only has a few points sparsely plotted throughout the US. Being separated, contiguity plays no part in the analysis. There are too few points to determine if there is any correlation between the data and any spatial attribute. This data should just be presented with some other widget that creates less distraction.

Adding more points, as in figure 1b, illustrates the problem of pattern recognition. People recognize patterns when none exist³. Due to this, math and the analyst should clearly communicate whether the data is spatially random or not. As a test, see if you can correctly guess which, if any, of the panels in figure 2 is an example of “Complete Spatial Randomness”. (Answer at end of ‘Conclusion’ section).

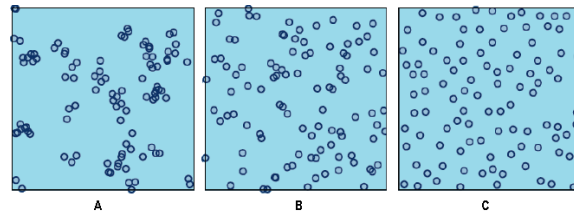


Figure2. Are any of these plots spatially random?

The tiniest leap of imagination is required to go from aerial imagery to a “standard” map. Distractions caused by a color raster image are replaced by vector based border lines and single colors for each geographic unit. The only thing gained by using such a map is added sense of size given by all the state borders. Producing such a map only takes a few lines of code:

```
PROC GMAP
  DATA=mapsgfk.us
    (WHERE=(statecode NOT IN
      ('AK', 'HI')))
  MAP=mapsgfk.us;
  ID id;
  CHORO state / NOLEGEND;
RUN;
QUIT;
```

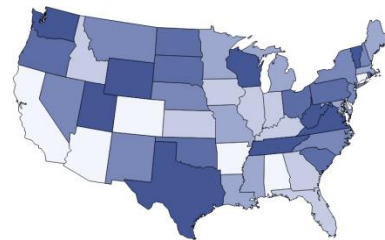


Figure 3. Map of the US with PROC GMAP

DON'T BE EVERYBODY

In an effort to extend our storytelling capabilities, this section will look at three less common geographic visualizations. Adjacency maps, process maps, and statistics are useful methods of getting across a message that would be hard to do with plain maps. While numbers are not a particularly graphical element, it is included as a reminder that sometimes the simple methods are the best methods.

ADJACENCY

Everything happens somewhere. Some things also happen between locations. North Carolina may be home to 100 service representatives, and South Carolina home to only 40. How much work is shared between the 2 states? By drawing lines from the center of each state to the centers of states next to them, an adjacency graph can be created. Edges are an intuitive representation of the flow, or shared contribution, between two connected nodes.

Figure 4, produced by example code 2, shows both different colors and line widths as methods of categorizing data values. Small gradations in line width are difficult to make out, so only a few can be used without sections getting too chunky looking. The graph is a more abstract shape than the map. Using the border of the US map helps retain a sense of place for the state centroids.

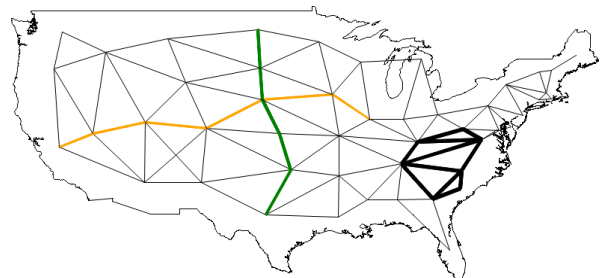


Figure 4. Adjacency graph of US states

PROCESS ATTRITION

The funnel diagram is a map used to show attrition through the steps of a process. When those steps occur in different places, putting data on a map again comes to mind. Figure 5 is Charles Minard's map⁴ of Napoleon's March on Moscow. This is a famous example showing the troop loss as Napoleon's army marched to Moscow in 1812. This is a very nice map, with a lot of detail.

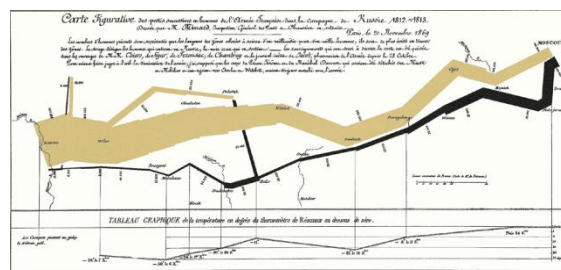


Figure 5. Napoleon's march on Moscow

When an extra week for one image is not available, a simpler method is called for. Figure 6 shows an adaptation of the previous figure in the style of a metro map⁵ is a good metaphor that can be produced much quicker, as in example code 3. Relative geographic relationships are preserved. Colors, labels, and symbols can be used to denote different stages/places and process attrition.

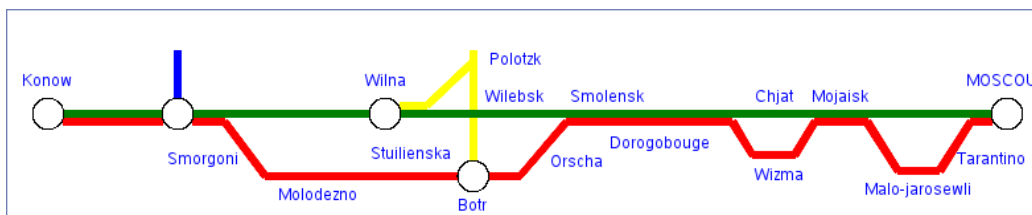


Figure 6. Metro map version of Napoleon's march on Moscow

MATH

No analysis is complete without a regression and p-values! An additional type of correlation can be tested when data has a location component. The null hypothesis of "spatial autocorrelation" is that there the spatial distribution of data is the result of a random process. If there is a non-random process, it could be a clustering force, causing values to be more similar the closer they are to each other. There could also be a dispersive force at play, causing like values to be farther away from each other⁶.

One such test for "Spatial Autocorrelation" is Moran's I. This test results in the eponymous Moran's I statistic. This value can be converted into a z-score and then again into a p-value. Chart 1 can be used to help Interpret these values.

z-score (Standard Deviations)	p-value (Probability)	Confidence level
< -1.65 or > +1.65	< 0.10	90%
< -1.96 or > +1.96	< 0.05	95%
< -2.58 or > +2.58	< 0.01	99%
P is not statistically significant: Accept H0		
P is statistically significant, z-score is positive: Reject H0, clustering effect		

P is statistically significant, z-score is negative: Reject H0, dispersive effect

Table 1. How to interpret the results of Moran's I

Example code 4 implements the Moran's I test. Using the population by county data helpfully provided in the MAPS library, this test can be ran on each state. For North Carolina, the I-statistic comes out to 0.0816 with a z-score of 1.454. Unable to reject the H0, there appears to be no spatial process at work for the distribution of North Carolinians in the year 2000, at least at the county aggregation level.

Figure 7, produced by code example 5, shows the five counties with the highest (green) and lowest (light pink) populations. High population counties appear clustered in the center of the state. Low population counties are split between the far ends of the state. To the untrained eye, this might appear to be a real clustering effect. Maps do not always tell the truth, that is why they should not always be used.

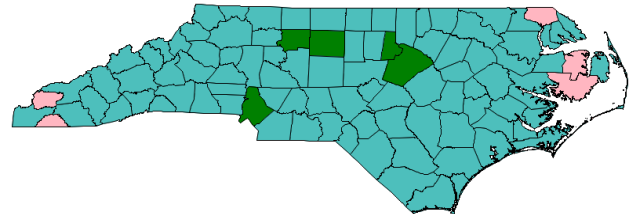


Figure 7. High & low NC county populations.

CONCLUSION

SAS® has many different ways to do the same thing, and this is true with creating graphics. Mastering the vagaries of the annotate data set and SAS/GRAPH functions can take time. The Graph Template Language also allows for custom drawing ability. At the expense of some freedom, the tedium of getting pixel perfect art can be avoided by using Visual Analytics.

Just because it has map in the procedure name, PROC GMAP is not limited to just creating generic maps. It is possible to create many different visual metaphors using the underlying features of the data. While the map libraries provide an excellent set of shape files to work with, they are not always required. Geographic features such as area and adjacency can easily be extracted and represented with bar, line, or other types of graphs. Hopefully this paper will inspire more creative ideas when it comes to visualizing and communicating location data.

The answer is: B. Panel 'A' shows a clustering force at work. Panel 'C' shows a dispersive effect

EXAMPLE CODE

Sample 1: Using an image and additional annotation

```
DATA annotate_image;
  xsys='2'; ysys='2'; hsys='3'; when='A'; function='image';
  imgpath='/home/SESUG2017/satellite_usa.jpg'; style='fit';
  x=380; y=190; /* x,y = numbers of pixels in image */
RUN;

DATA annotate_points;
  RETAIN xsys ysys '2' hsys '3' when 'A' function style text position size;
  function='label'; style="Ariel"; text='Data'; position='3'; size = 5;
  /* ... set 'color=; x=; y=; OUTPUT;' for each text label */
  function='symbol'; style='marker'; text='V'; position='5'; size = 4;
  /* ... set 'color=; x=; y=; OUTPUT;' for each star marker */
RUN;

DATA border_map; /*Make a border equal to size of the image*/
  INPUT id x y @@;
  DATALINES;
```

```
1 0 0 1 380 0 1 380 190 1 0 190
```

```
;
```

```
RUN;
```

```
PATTERN v=e;
```

```
PROC GMAP MAP=border_map ANNO=annotate_image;
```

```
    ID id;
```

```
    CHORO id / NOLEGEND ANNOTATE= annotate_points;
```

```
RUN; QUIT;
```

Sample 2: Converting polygons to an adjacency graph

```
DATA lower48;
```

```
    SET mapsgfk.us_states(WHERE=(statecode NOT IN ('AK','HI')));
```

```
    region = 1;
```

```
RUN;
```

```
PROC GREMOVE DATA=lower48 OUT=outline48; /* Create frame for graph */
```

```
    BY region;
```

```
    ID state;
```

```
RUN;
```

```
%annomac; /* Required to initialize annotation macros */
```

```
%centroid(lower48, state_center, state); /* Find the center of each state */
```

```
PROC SQL; /* Cartesian join =  $\otimes$ . Use bounding boxes to improve performance */
```

```
    CREATE TABLE adjacency_list AS /* Prevent dups, lower value goes first */
```

```
    SELECT DISTINCT CASE WHEN (a.state < b.state) THEN a.state
                        ELSE b.state END AS statel,
                   CASE WHEN (a.state < b.state) THEN b.state
                        ELSE a.state END AS state2
```

```
    FROM lower48 AS a INNER JOIN
```

```
        lower48 AS b
```

```
    ON a.x = b.x AND
```

```
        a.y = b.y
```

```
    WHERE a.state NE b.state;
```

```
QUIT;
```

```
PROC SQL; /* Add x, y coordinates for both states onto each row */
```

```
    CREATE TABLE adjacency_list2 AS
```

```
    SELECT c.statel, c.x1, c.y1, c.state2, d.x AS x2, d.y AS y2
```

```
    FROM (SELECT a.statel, a.state2, b.x AS x1, b.y AS y1
```

```
            FROM adjacency_list AS a INNER JOIN
```

```
                state_center AS b
```

```
            ON a.statel = b.state) AS c INNER JOIN
```

```
        state_center AS d
```

```
    ON c.state2 = d.state;
```

```
QUIT;
```

```
DATA state_graph (DROP=statel state2 x1 x2 y1 y2);
```

```
    SET adjacency_list2;
```

```
    FORMAT function $4. color $6.;
```

```
    RETAIN when 'A' xsys ysys '2' color 'black' line 1 size 2;
```

```
    function = 'move'; x = x1; y = y1; OUTPUT;
```

```
    function = 'draw';
```

```
    IF Statel IN (13, 21, 37, 45, 47, 51, 54) AND
```

```
        state2 IN (13, 21, 37, 45, 47, 51, 54) THEN size = 4;
```

```

ELSE size= 1;
IF statel IN (6, 8, 17, 19, 31, 32, 49) AND
state2 IN (6, 8, 17, 19, 31, 32, 49) THEN DO;
color = 'orange'; size = 2;
END;
ELSE IF statel IN (20, 31, 38, 40, 46, 48) AND
state2 IN (20, 31, 38, 40, 46, 48) THEN DO;
color = 'green'; size = 3;
END;
ELSE color = 'black';
x = x2; y = y2; OUTPUT;
RUN;

```

```

GOPTIONS RESET=all CBACK=white CPATTERN=white;
PROC GMAP MAP=outline48;
ID region;
CHORO region / NOLEGEND ANNOTATE=state_graph;
RUN; QUIT;

```

Sample 3: Drawing a Subway Diagram

```

%MACRO circles(file_name, r, c_x, c_y); /* Save typing repeated circles */
DATA &file_name;
FORMAT function $8. x y best12. text $15. color $6.;
RETAIN xsys ysys '2' hsys '3' when 'A' position '5'
radius &r size line 1 text '';
deg2rad = 3.1415926535/180.0;
function = 'move'; x = &c_x; y = &c_y; OUTPUT;
DO degree = 0 TO 359 BY 1;
IF degree = 0 THEN DO;
function = 'poly';
style = 'empty';
END;
ELSE DO;
function = 'polycont';
color = 'black';
END;
x = &r * COS(degree*deg2rad) + &c_x;
y = &r * SIN(degree*deg2rad) + &c_y;
OUTPUT;
END;
RUN;
%MEND circles;

%circles(c1, 1.5, 4.0, 10.0); /*Kowno*/
%circles(c2, 1.5, 16.5, 10.0);
%circles(c3, 1.5, 36.5, 10.0); /*Wilna*/
%circles(c4, 1.5, 96.5, 10.0); /*MOSCOU*/
%circles(c5, 1.5, 45.0, 4.0); /*Botr*/

DATA non_circles;
FORMAT function $8. x y best12. size 1. text $15. color $6.;
RETAIN xsys ysys '2' hsys '3' when 'A' position '5'
line 1 size color function style;
size = 1;
color = 'yellow'; /* Yellow is overlapped by green, so goes first */
function = 'move'; x = 38.01; y = 10.75; OUTPUT;

```

```

function = 'draw'; x = 40.5; y = 10.75; OUTPUT;
                x = 45.0; y = 15.0; OUTPUT;
function = 'move'; x = 45.0; y = 16.0; OUTPUT;
function = 'draw'; x = 45.0; y = 5.5; OUTPUT;
color = 'blue';
function = 'move'; x = 16.5; y = 16.0; OUTPUT;
function = 'draw'; x = 16.5; y = 11.5; OUTPUT;
/* ... drawing instructions cut for space. Contact me if interested. */

function = 'label'; style = "'Albany AMT'"; color = 'blue'; size = 2;
text = 'Konow'; x = 4.0; y = 13.5; OUTPUT;
text = 'Smorgoni'; x = 19.0; y = 6.0; OUTPUT;
/* ... text labels cut for space. Contact me if interested. */
RUN;

DATA march;
SET non_circles c1 c2 c3 c4 c5;
RUN;

DATA border_map;
INPUT id x y @@;
DATALINES;
1 0 0 1 100 0 1 100 20 1 0 20
;
run;

PATTERN v=e;
PROC GMAP MAP=border_map ANNO=march;
ID id;
CHORO id / NOLEGEND;
RUN;
QUIT;

```

Code 4: Calculating Moran's I, a Measure of Spatial Autocorrelation.

```

DATA nc_map; /*Create Map*/
SET mapsgfk.us_counties (WHERE=(statecode='NC'));
RUN;

PROC SQL; /*Create data set*/
CREATE TABLE nc_pop AS
SELECT DISTINCT b.county, SUM(a.pop) AS pop2000
FROM maps.uscity AS a INNER JOIN
(SELECT DISTINCT state, county
FROM nc_map) AS b
ON a.state = b.state AND
a.county = b.county
GROUP BY a.county;
QUIT;

PROC SQL; /*Create list of joins between county polygons*/
CREATE TABLE temp AS
SELECT a.x, a.y, b.county AS p
FROM (SELECT DISTINCT x, y FROM nc_map
GROUP BY x, y HAVING COUNT(*) GT 1) AS a INNER JOIN
nc_map AS b
ON a.x = b.x AND

```

```

        a.y = b.y
ORDER BY a.x, a.y;

CREATE TABLE join_list_unique AS /*list of unique joins*/
SELECT DISTINCT
        CASE WHEN one.p < two.p THEN one.p ELSE two.p END AS county1,
        CASE WHEN one.p < two.p THEN two.p ELSE one.p END AS county2
FROM temp AS one INNER JOIN temp AS two
ON one.x = two.x AND
    one.y = two.y
WHERE one.p NE two.p
ORDER BY CALCULATED county1, CALCULATED county2;

CREATE TABLE join_list_all AS /*count of joins for each polygon*/
SELECT DISTINCT county1, COUNT(*) AS counts
FROM (SELECT DISTINCT one.p AS county1, two.p AS county2
      FROM temp AS one INNER JOIN temp AS two
      ON one.x = two.x AND
          one.y = two.y
      WHERE one.p NE two.p)
GROUP BY 1
ORDER BY county1;
QUIT;

PROC SQL; /* Create values to calculate Moran's I */
SELECT COUNT(*), SUM(pop2000)/COUNT(*) INTO :counties, :pop_avg
FROM nc_pop;
QUIT;

PROC SQL;
CREATE TABLE temp AS
SELECT a.county1, (a.counts ** 2) AS counts_sq,
        (b.pop2000 - &pop_avg) ** 2 AS delta_sq
FROM join_list_all AS a INNER JOIN nc_pop AS b
ON a.county1 = b.county;

SELECT SUM(counts_sq), SUM(delta_sq) INTO :joins_sq, :delta1_sq
FROM temp;
QUIT;

PROC SQL;
CREATE TABLE temp AS
SELECT c.county1, c.county2, c.xi, d.pop2000 AS xj
FROM (SELECT a.county1, a.county2, b.pop2000 AS xi
      FROM join_list_unique AS a INNER JOIN
          nc_pop AS b
      ON a.county1 = b.county) AS c INNER JOIN
          nc_pop AS d
ON c.county2 = d.county;

SELECT COUNT(*), SUM((xi - &pop_avg)*(xj - &pop_avg)) INTO :joins, :delta2
FROM temp;
QUIT;

DATA morans_i;
E = -1 / (&counties - 1);
sigma = sqrt(((&counties ** 2 * &joins) +

```

```

        (3 * &joins ** 2) -
        (&counties * &joins_sq)) /
        (&joins ** 2 * (&counties ** 2 - 1)));
    morans_i = (&counties * &delta2) / (&joins * &delta1_sq);
    z = (morans_i - E) / sigma;
RUN;

```

Code 5: 5 highest and lowest population counties in NC, according to 2000 census data.

```

DATA nc_map; /*Create Map*/
    SET mapsqfk.us_counties (WHERE=(statecode='NC'));
RUN;

PROC SQL; /*Create data set*/
    CREATE TABLE nc_pop AS
    SELECT DISTINCT b.county, SUM(a.pop) AS pop2000
    FROM maps.uscity AS a INNER JOIN
        (SELECT DISTINCT state, county
         FROM nc_map) AS b
    ON a.state = b.state AND
        a.county = b.county
    GROUP BY a.county
    ORDER BY pop2000;
QUIT;

DATA nc_pop;
    SET nc_pop;
    IF _N_ <= 5 then pop_rank = 1;
    ELSE IF _N_ < 94 THEN pop_rank = 2;
    ELSE pop_rank = 3;
RUN;

PATTERN1 c=green;
PATTERN2 c=bibg;
PATTERN3 c=lightPink;
PROC GMAP MAP=nc_map DATA=nc_pop;
    ID county;
    CHORO pop_rank;
RUN;
QUIT;

```

REFERENCES

1. Google. Search for “Satellite image of the US”. Accessed September 13, 2017.
https://www.google.com/search?rlz=1C1GGRV_enUS758US758&q=satellite+image+of+the+us&oq=satellite+image+of+the+us&gs_l=psy-ab.3..0j0i22i30k1i3.6682.10276.0.10477.25.20.0.0.0.282.2132.0j10j3.13.0....0...1.1.64.psy-ab..12.13.2131...0i67k1j0i131k1.0.E8_qs775CFU
2. Dr. Robert Allison's SAS/GRAPH Examples. Link number 41. “Aligning Map and Image”. Accessed August 26, 2017.
http://robslink.com/SAS/democd41/peru_sat_info.htm
3. Greenberg, J. Koole, S. Pyszczynski, T. Dec 17, 2013 “Handbook of Experimental Existential Psychology”. Guilford Publications.
4. Edward Tufte, Store Poster: Napoleon's March, \$14.00. Accessed September 02, 2017.

<https://www.edwardtufte.com/tufte/posters>

5. Edward Tufte, Bulletin Board. “*Response to Block That Metaphor! The NASA Space Exploration Map*”. Accessed September 02, 2017.

https://www.edwardtufte.com/bboard/q-and-a-fetch-msg?msg_id=000012

6. ArcGIS Desktop. How Spatial Autocorrelation (Global Moran's I) works Accessed August 20, 2017.

<http://pro.arcgis.com/en/pro-app/tool-reference/spatial-statistics/h-how-spatial-autocorrelation-moran-s-i-spatial-st.htm>

ACKNOWLEDGMENTS

I would like to thank Barbara Okerson, RV/JMP Section Chair for accepting my abstract and paper; Jason Brinkley SESUG 2017 Academic Chair, Meenal Sinah, SESUG 2017 Operations Chair, members of the South East SAS Users Group (SESUG) Executive board, and SAS Institute for organizing and supporting this conference. I would also like to thank Professor Ranga Myneni of Boston University, without whose generosity and support, I would still think geography was mostly about knowing state capitals.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Seth Hoffman
GEICO
Seth.w.hoffman@gmail.com

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.

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