

Visual Analytics for Dynamically conditioned Choropleth Maps: QQplots, Scatterplot Smoothes and Two-way Tables

Chunling Zhang, (George Mason University), czhang1@gmu.edu

Yaru Li, (George Mason University), yli6@gmu.edu

Daniel Carr, (George Mason University), dcarr@gmu.edu

Abstract

The visual analytics presented in this paper augment conditioned choropleth maps. In the conditioned map, dynamic sliders partition the map into a 3 x 3 grid of partial maps. Two different variables are attached to the two partitioning sliders. One slider controls row membership in the grid and the other controls column membership. The analyst's visual impression and comparison of the partial maps can be made more quantitative by showing other analytics. The analytics described in this paper are modifications of conventional QQplots, smoothed scatterplots, and two-way tables of means, effects, and model statistics. One modification involves the use of weights. Most modifications speed the response in order to keep up with the dynamic partitioning sliders. For example, the smoothing widgets include the option to use an intermediate binning step when thousands of regions are involved. The talk provides live examples. The applications involve different kinds of region such as county elementary school districts, hexagon grids for three states, and nations of the world.

Keywords: CCmaps, QQplots, Scatterplots, Loess Regression, Binning, Two-way Tables, Weighted Smooth.

1.1 Introduction of CCmaps

Conditioned choropleth maps (CCmaps) is a dynamic Java shareware application for exploratory analysis of geo-spatially-indexed data. The goals of CCmaps include better-focused hypothesis generation about geo-spatial patterns. By providing dynamic partitioning sliders based on three variables, CCmaps enables an analyst to define what is meant by low, medium and high values. Stratification into three classes based on two conditional variables yields a 3 x 3 layout of partial maps. The maps are colored by the stratified depend variable. In CCmaps, user can choose the variables to be analyzed and displayed. The functions of CCmaps include dynamic conditioned maps, dynamic computation of two-way tables of means, effects, and model statistics for hypothesis test, dynamic conditioned scatterplots of the dependent variable and third risk factor, dynamic scatterplot smooth and dynamic conditional QQplots. The figure 1.1 displays the above aspects of CCmaps.

The CCmaps software is available via www.galaxy.gmu.edu/dcarr/ccmaps.

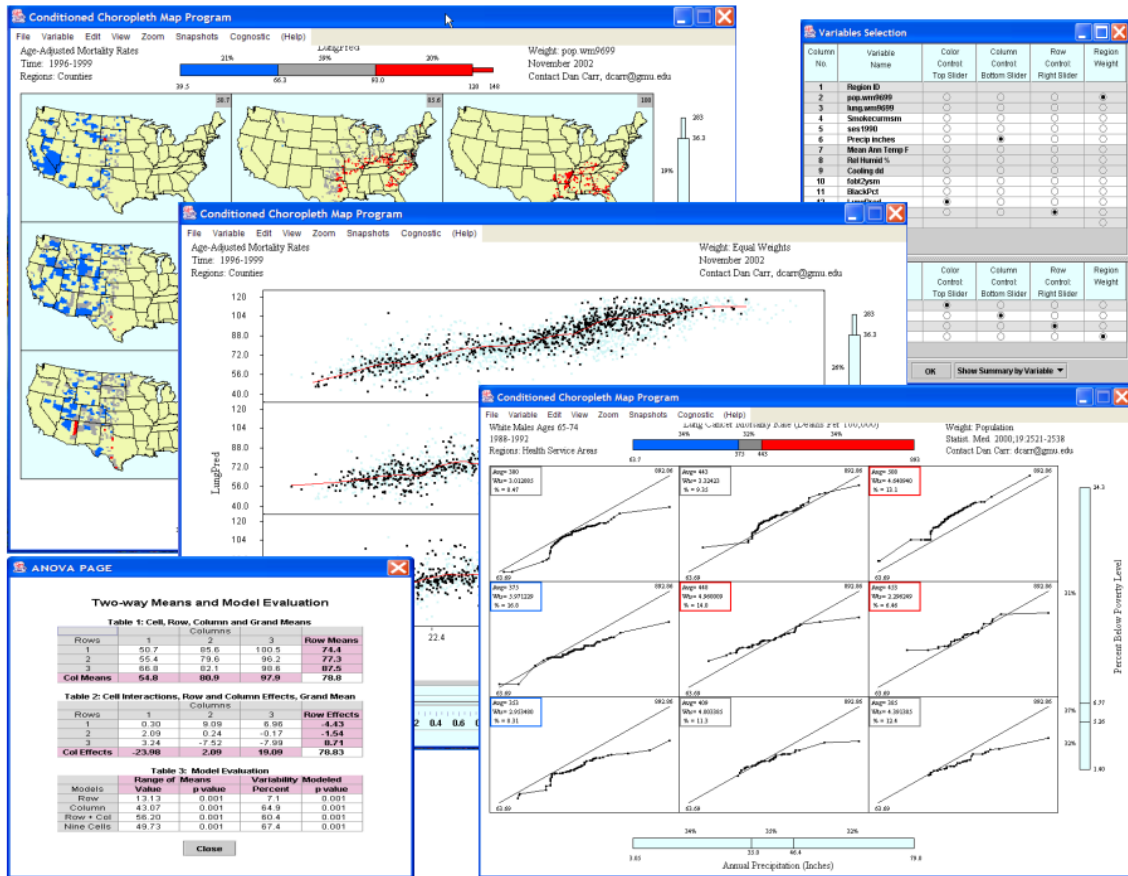


Figure 1.1 Overview of CCmaps

1.2 The contents of the paper

In section 2, we will introduce the Loess regression model, its parameters, and its algorithm for two-dimension dataset. In section 3, we will explain the binning method, which we use to speed up the process of scatterplot smoothing. In section 4, we take a glance of a parameter, smooth resolution which is for adjusting the curve of scatterplot smoothing. In section 5, we give a view of dynamic weighted QQplot in CCmaps. In section 6, we give the illustrative explanation of dynamic two-way table for statistical description and promoting interactive hypothesis generation.

2. Loess Smooth

CCmaps provides the conditioned dynamic scatterplot, (see the left graph in Figure 2.1), also it gives the dynamic smoothing of the scatterplot (see the right graph in Figure 2.1).

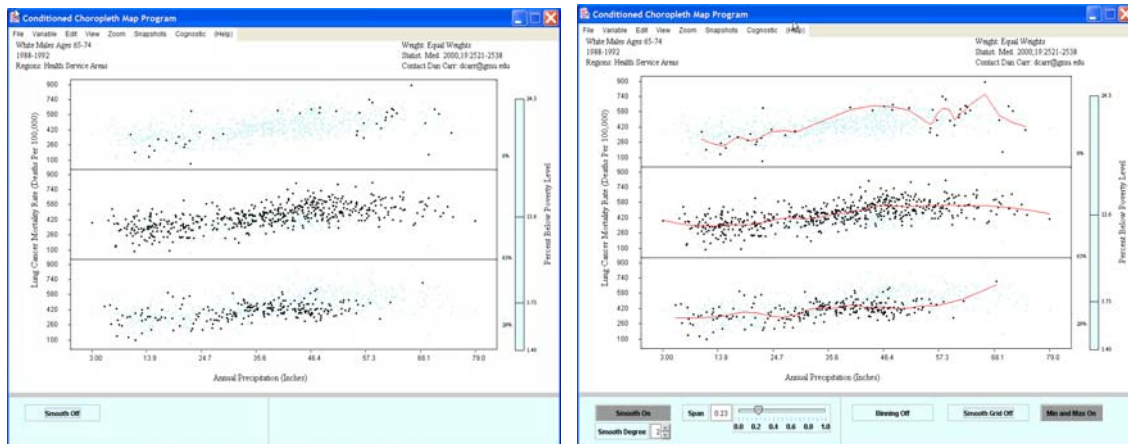


Figure 2.1 Scatterplot and Loess Smooth in CCmaps

There are lots of methods for smoothing. For example, smoothing splines, local regression with knot selection, wavelets, various kernel methods, and various global regressions. In CCmaps, the smoothing method used is LOESS, also called locally weighted polynomial regression. This method was proposed by Cleveland (1979) and further developed by Cleveland and Devlin (1988).

We will explain Loess model, its parameters and its algorithm. Because in CCmaps, we aim to smooth the scatterplot in CCmaps, in the following, all the details in Loess are for two-dimension data.

The model of Loess:

$$y_i = f(x_i) + \varepsilon_i \quad i = 1, \dots, n,$$

where y_i are observations of a response, x_i are observations of independent variable. ε_i are random variables. f is a function which can be approximated locally by a class of functions, usually a polynomials a certain degree $\{f_{i,k}(x)\}$, where $f_{i,k}$ are k degree of local polynomial functions related to point x_i .

The idea of Loess is weighted least square polynomial fitting. That is, for fitting a point x , define a neighborhood of x . the f within its neighborhood is estimated by using the points in the range of this neighborhood. In doing this, a weight function, $w(z)$ is incorporated by giving greater weight to the x_i in the neighborhood that are closer to x and less weight to those further points. In CCmaps, we use tri-cubic weighting function with the following formula:

$$w(z) = \begin{cases} (1-|z|^3)^3 & \text{for } |z| < 1 \\ 0 & \text{others} \end{cases}$$

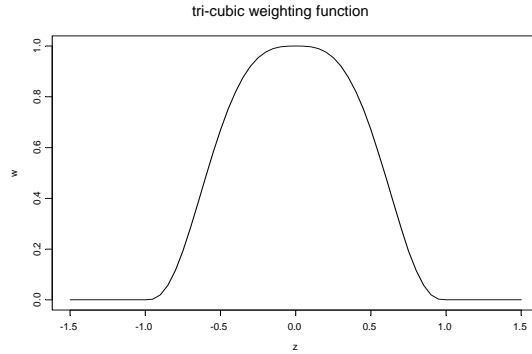


Figure 2.2 Tri-Cubic weighting function $w(z)$

In the local neighborhood, we usually assume that y_i approximately follow Gaussian distribution with constant variance. That is, we have that following model:

$$y_i = f_{i,k}(x_i) + \varepsilon_i \quad i=1, \dots, m,$$

Where m is points in the neighborhood, y_i and x_i are the observed values of response and independent variables respectively. ε_i are Gaussian random variables with mean of zero and constant variance. So the $f_{i,k}$ (or the coefficients of $f_{i,k}$) can be estimated based on least squares, that is, to minimize

$$\sum_{j=1}^m w((x_j - x_i)/h) (y_j - f_{i,k}(x_j))^2 \quad (*)$$

Where h is width of the neighborhood of x . for example, if $k=2$, we need to minimize

$$\sum_{j=1}^m w((x_j - x_i)/h) (y_j - a_0 - a_1(x_j - x_i) - a_2(x_j - x_i)^2)^2$$

by solving the least square functions, we estimate the coefficients a_0 , a_1 , and a_2 , thus, we can estimate $f_{i,k}$. It is easy to see that when k is zero, the loess becomes weighting average moving regression.

Parameters in Loess:

There are many possible choices for the parameter bandwidth h in the formula (*), Cleveland and Loader (1996) mentioned three main types of h . Fixed bandwidth, nearest-neighbor bandwidth, and the adaptive bandwidth, which is the combination of the former two types. In CCmaps, we use nearest-neighbor bandwidth.

To explain the nearest-neighbor bandwidth, we will introduce an important parameter α called *span*, also called smoothing parameter. Span α is a number in the range of 0 and 1 and represents the portion of points in the neighborhood of a fitted point. That is, the number of points in the neighborhood, m is obtained by rounding up the multiplication of n and α (where n is the total points number in the dataset). The bandwidth h for point x is the distance from x to the m th closest x_i . Easy to see, when α is smaller, less points are

involved in fitting a point, thus more local information is reflected in the line. Contrastingly, bigger span reflect more global information and makes the line smoother. The bigger span can smooth out the outliers, but if span is too big, some important local information may get lost. CCmaps provides user the flexibility to get an appropriate span by dynamically adjusting the slider bar (see the right graph in Figure 2.3)

Besides span α , degree k , the degree of a polynomial fitting function in the regression model is also an important parameter in Loess. CCmaps provides four choices of degree, which are 0,1,2 and 3 (see the right graph in Figure 2.3).

Algorithm (for span α , degree k)

To fit the values for points $x_i, i = 1, \dots, n$; where n is the number of different x coordinates in the dataset.

1. Compute m , the number of points in the neighborhood of the fitted point x_i , $m = n * \alpha$.
2. Select the m closest points from x_i ($x_j, j = 1, \dots, m$) by sorting the distances of points from x_i
3. Compute h , the maximum distance of the m selected points from x_i , $h = \max \{|x_j - x_i|\}$
4. Compute the weights of the m points in the neighborhood of x_i . $w(x_j) = (1 - (|x_j - x_i| / h)^3)^3, j = 1, \dots, m$.
5. Set temporary degree $d_0 = k$
6. The weights gotten in step 4 are used to perform a weighted least squares fit of the local mode in the neighborhood of x_i .
7. If the matrix of the least square equation is singular, $d_0 = d_0 - 1$, return to step 5; else, Compute all the $d_0 + 1$ coefficients of d_0 degree of polynomial in the model by solving the least square equations.
8. Compute the value of the solved polynomial at x_i .

Splus, R and SAS also provide the Loess routine. In CCmaps, we code in java to implement the Loess smooth instead of calling the available routine. The comparison of some results from our program and Splus shows that although in some situations there are some small discrepancies between these them, usually the two results are very close or identical. Two examples are given in the Figure 2.2. (Note that red line represents the result from CCmaps and black one represents that from Splus.)

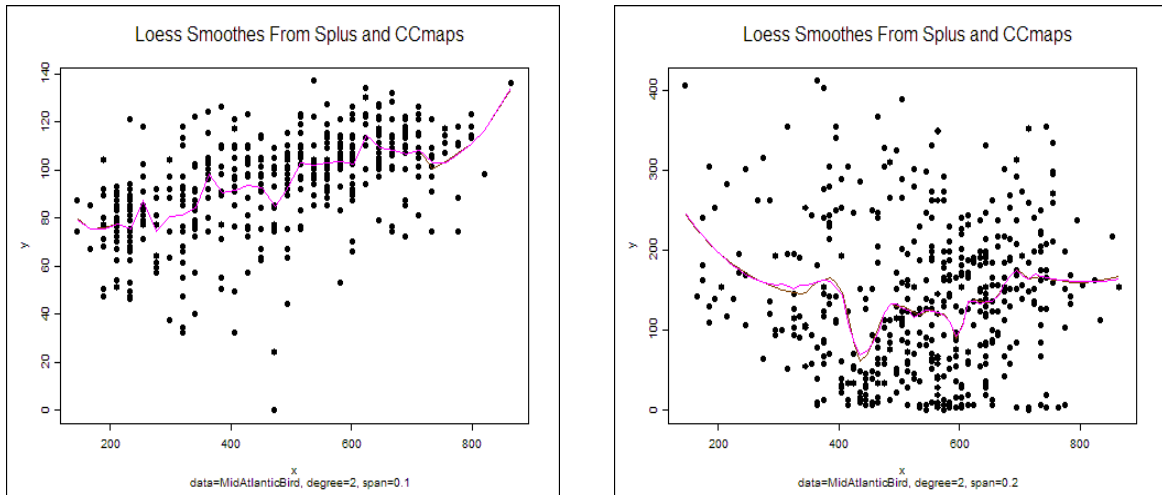


Figure 2.3 comparisons of smoothing results in CCmaps and Splus

CCmaps provides two choices to weight each point. The default setting is equal weight for each point; CCmaps also allows user to choose the values of any variable to weight points. For example, in the situation of analyzing the county data, the population in each county seems a more appropriate weight for each point than equal weight does (see left graph in Figure 2.3).

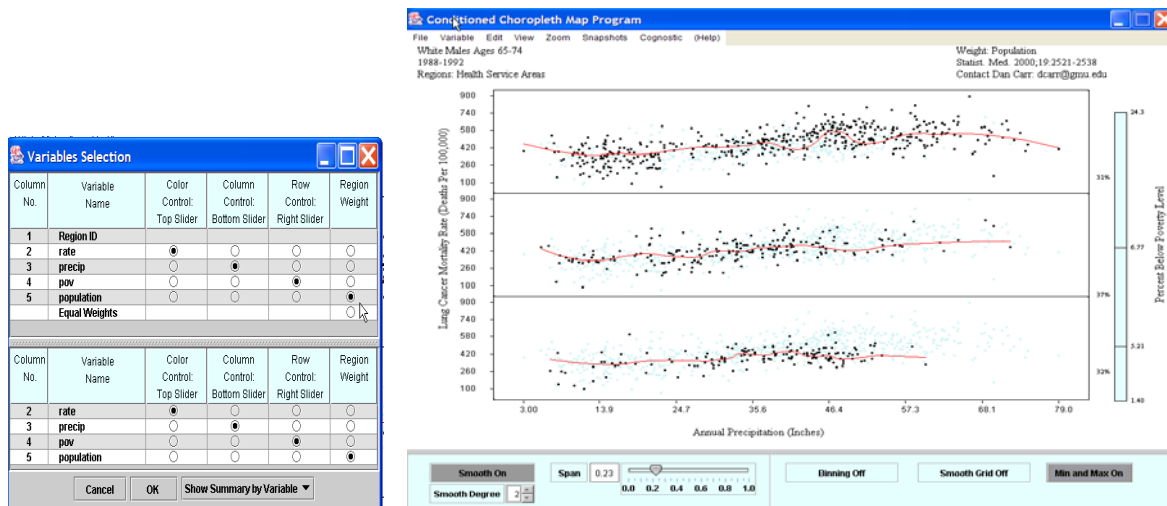


Figure 2.4 Weight choices for each point in smoothing procedure

1. Binning in Scatteplot Smooth

In Loess, the default points to be fitted are the different x coordinates of the points in the dataset. From the algorithm of Loess, we can see that when there is large number of different x coordinates, the process of Loess may take a quite long time, and especially when degree is large because of a matrix equation solved. In CCmaps, the dynamic display

demands even faster computation. To speed up the smooth, we employ the idea of binning to aggregate the large set of points into a small one before smoothing. (Carr, D. B. (1991) gave a better understanding of binning in both computation and visualization.) The following are the process of binning.

1. Partition the point space into cells.
2. Aggregate all the points in a cell into a single point by doing the following computation.
 - The coordinate of the aggregated point is the weighted average of the coordinates of all the points in the cell.
 - The weight of the aggregated point is the summation of the weights of all the points in the cell.

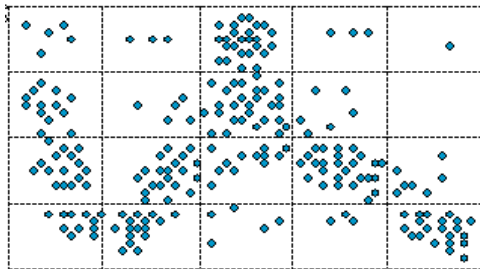


Figure 2.4 Partitioning data into grids

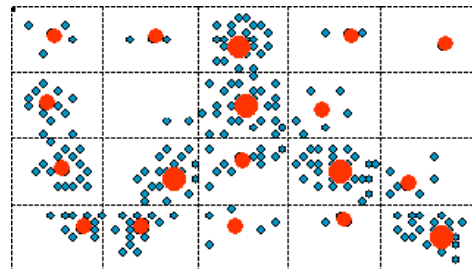


Figure 2.5 Aggregation of large data set

About binning, one thing we need to notice is that if the data set is large and the data are randomly distributed, binning process speeds up the smooth process. However, if the data has few number of x values as shown in Figure 2.6, the binning process may slow down the following smoothing process because the binning process may produce more x values.

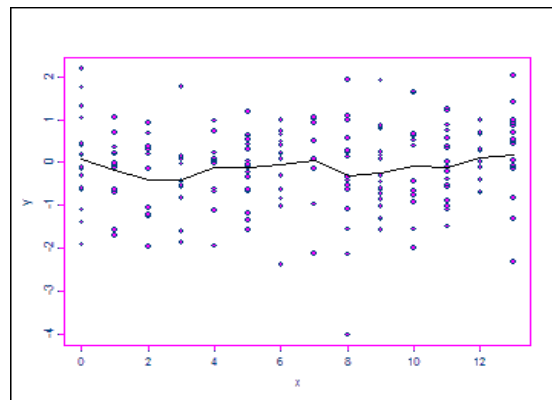


Figure 2.6 Few x values in data set

CCmaps shows that binning process speed up the smoothing while doesn't change much of the original smoothing curve (without binning). See the example in Figure 2.7.

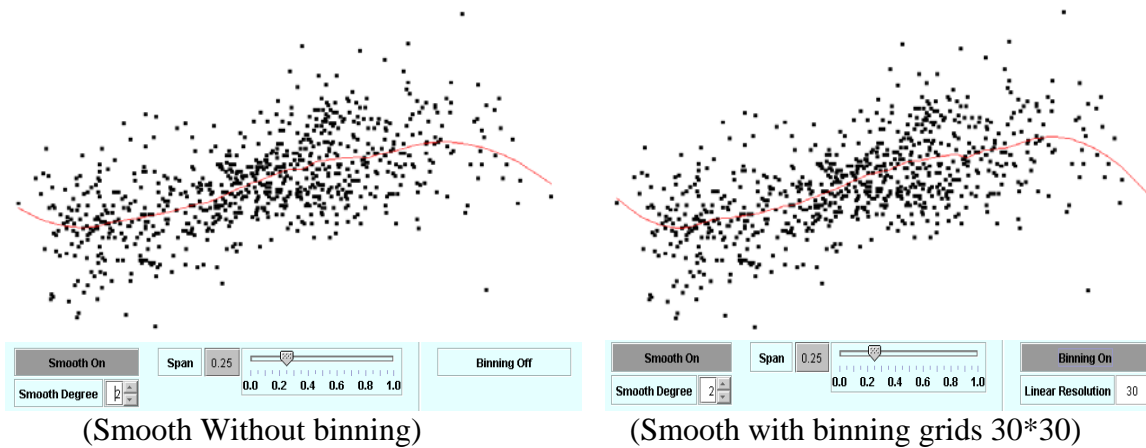


Figure 2.7

4. Smooth Resolution

Usually, the more points are fitted, the smoother a line is. However, this smoother line is at the cost of slowing down the smoothing speed because more computation is involved. To allow a tradeoff between smoothness and speed, we modify the Loess algorithm by providing user an option for setting the number of equally separated points to be fitted. See the following figures 4.1, 4.2 and 4.3 respectively show 5, 200 and the default numbers of points are fitted.

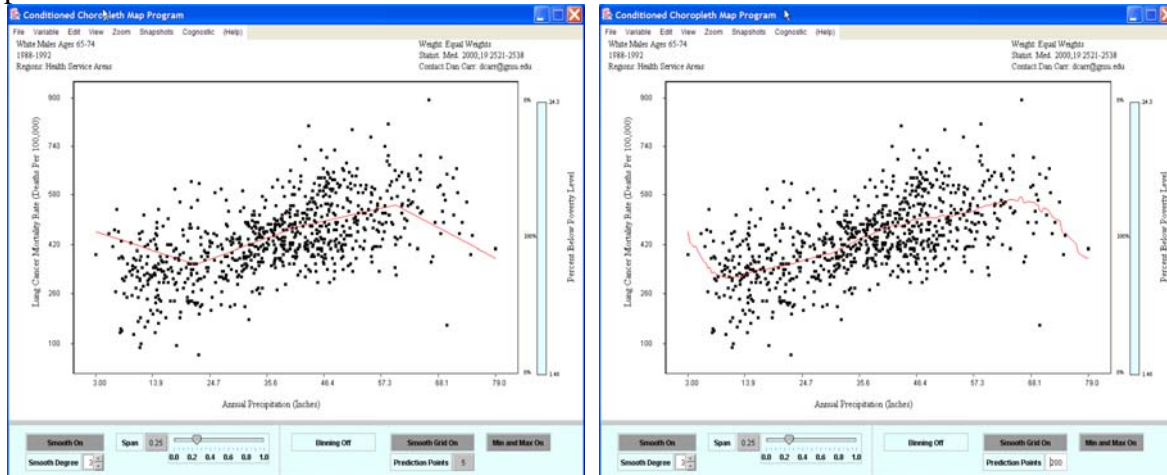


Figure 4.1 5 equally separated points are fitted Figure 4.2 200 equally separated points are fitted

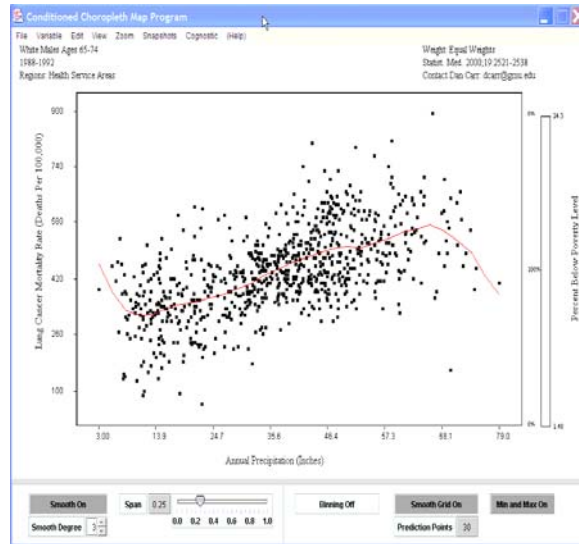


Figure 4.3 Default setting for the number of points to be fitted

5. Dynamic QQ-plot in CCmaps

CCmaps provides weighted QQ-plots for comparing panel values with the composite of values from other panels, that is, the quartiles of the dependent variable in each panel vs. those in all other panels. Cleveland (1993) provides a good interpretation of QQ-plot. In each panel of the 3 x 3 layout of QQ-plots, CCmaps also shows cumulative weights and the corresponding percent of the population for individual panels. The QQ-plots and statistics dynamically update with the movement of the partitioning sliders.

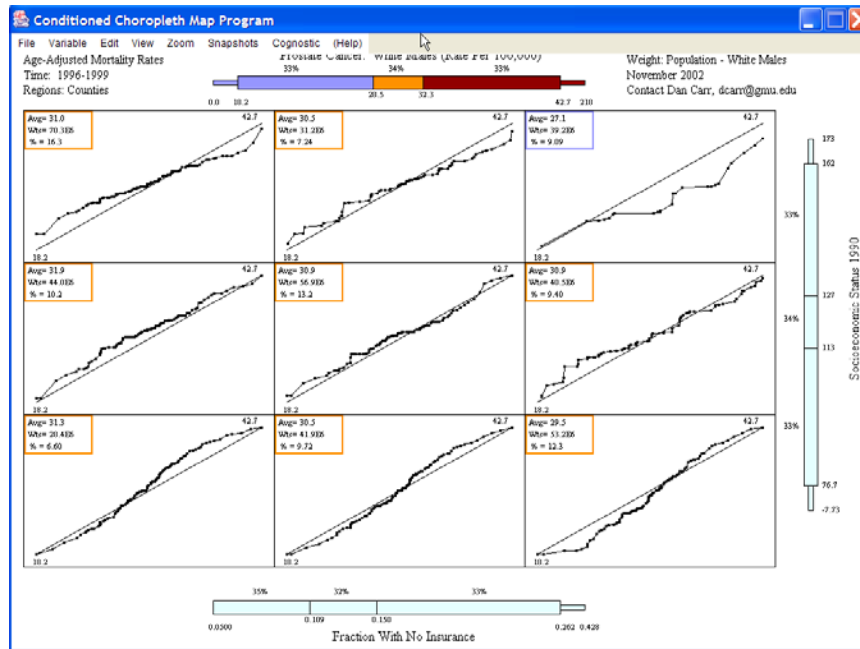


Figure 5.1: dynamic QQ-plot

6. Two-way tables

Two-way table is one of new visual analytical features that have been added to CCmaps recently. It is used to assess and compare simple models that relate two explanatory variables to a dependent variable. Basically, they are analysis of variance models. The goal of two-way tables is to provide descriptive statistic about the data and thus promote interactive hypothesis generation.

CCmaps provides three two-way tables of means and main effects: (1) cell, row, column and grand means; (2) cell interactions, row and column effects, grand mean, (3) model evaluation. Figure 6.1 is an example.

Table 1: cell, row, column and grand means

In Figure 1, the conditioning variables are annual precipitation and the percent below the poverty level. Two dynamical sliders partition the distribution of mortality rates into 3x3 groups. Table 1 provides cell means (average mortality rates) for 9 groups, row means for 3 rows, column means for 3 columns and the grand mean. All values for cells and margin are calculated as *population-weighted* means.

Table 2: cell interactions, row and column effects, grand mean

Table 2 comes directly from Table 1. It provides cell interactions, row and column effects, and also the grand mean. To obtain these statistics, the algorithm needs to accommodate the case where there are missing cells in Table 1, and incorporate weights as well. We use the following straightforward calculations to get the data:

$$\begin{aligned} \text{Cell value} &= \text{cell mean} - \text{row mean} - \text{column mean} + \text{grand mean} \\ \text{Row effects} &= \text{row mean} - \text{grand mean} \\ \text{Column effects} &= \text{column mean} - \text{grand mean} \end{aligned}$$

Table 3: model evaluation

Table 3 is model evaluation. Four models are considered in table 3: (1) row effect only; (2) column effects only; (3) additive row and column effect; (4) two-way interaction. There should be an option to set the number of permutations, but maybe later. I probably will want to add some text, perhaps in a help file accessible from this page..

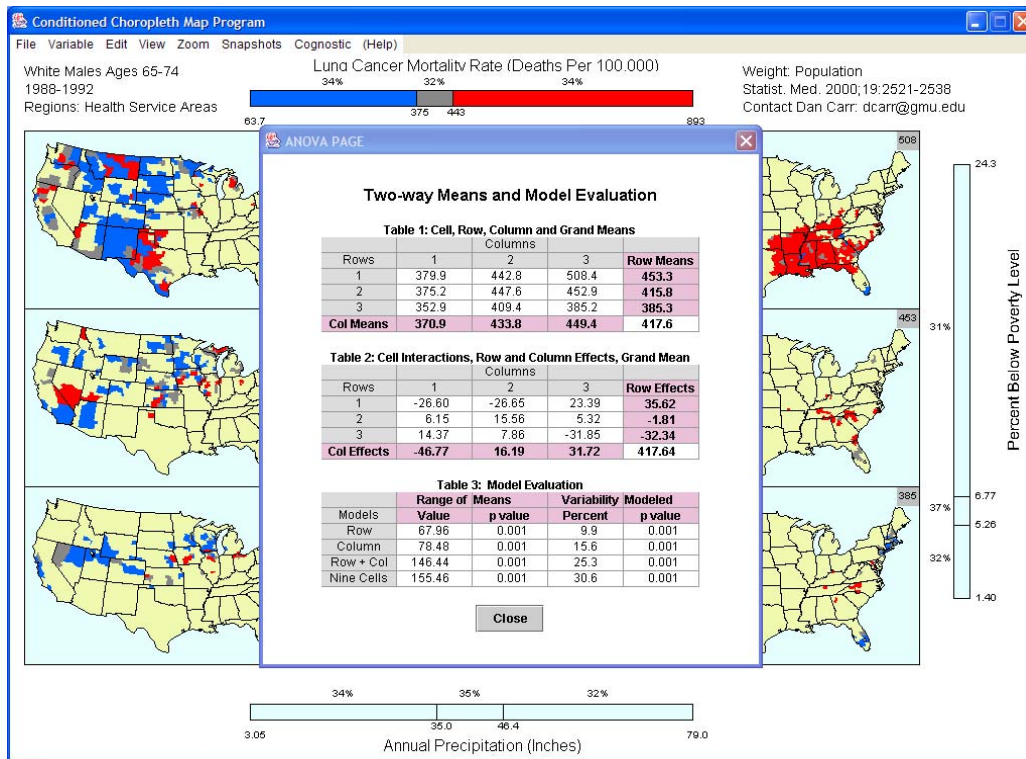


Figure 6.1: Two-way means and model evaluation

For each model there are two summary statistics: one is the range of means, the other is r-squared value or variability percent. For the row model, the range of means is the maximum minus the minimum of the row means in Table 2. The calculations are similar for the other models. The range of means can be assessed in terms of practical significance.

If it is not of practical significance, then there is little reason to consider the model any further.

R-Squared is the percent of variation about the grand mean explained by the model. The Row model uses the mean for each row to model the region values for the region in each row. The Column model is analogous. The interaction model uses the cell means from Table 1 to model the region values for regions belong to the cell. A small value R-squared for a model indicates that much variation remains to be explained. A large R-squared is a good sign but does not preclude confounding with other variables.

In addition to two summary statistics, a permutation test is performed to get the p-value, respectively. The p-value indicates the occurrences fraction of values as extreme when region class memberships are permuted. The default is 1000 permutations. A small p-value suggests that the observed magnitude is not likely to be based on a random association between the conditioning variable(s) with the dependent variable.

Reference:

Carr, D. B., Littlefield, R. J., Nicholson, W. L. and Littlefield, J. S. (1987). Scatterplot matrix techniques for large N. *Journal American Statistical Association* **83**, 424-436.

Carr, D. B. (1991). Looking at large data sets using binned data plots. In *Computing and Graphics in Statistics*. A. Buja and P. Tukey, eds. Springer-Verlag, New York. pp. 7-39.

Cleveland, W.S. (1979) "Robust Locally Weighted Regression and Smoothing Scatterplots," *Journal of the American Statistical Association*, Vol. 74, pp. 829-836.

Cleveland, W.S. and Devlin, S.J. (1988) "Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting," *Journal of the American Statistical Association*, Vol. 83, pp. 596-610.

Chambers, J.M., and Hastie, T.J. (1991). *Statistical Models in S*, 309-376.

Cleveland, W.S., and Grosse, E. (1991) Computational Methods for Local Regression. *Statistics and Computing*, Vol. 1.

Cleveland, W. S. (1993) *Visualizing Data*, Hobart Press, Summit, New Jersey.

Cleveland, W.S. and Loader, C.L. (1996). **Smoothing by Local Regression: Principles and Methods**. In W. Härdle and M. G. Schimek, editors, *Statistical Theory and Computational Aspects of Smoothing*, pages 10-49. Springer, New York.

Daniel B. Carr, Yuguang Zhang, Yaru Li, George Dynamically Conditioned Choropleth Maps: Shareware For Hypothesis Generation and Education.
<http://www.public.iastate.edu/~dicook/scgn/v132.pdf>

Graybill, F.A. and Iyer, H.K. (1994) *Regression Analysis: Concepts and Applications*, Duxbury Press, Belmont, California.

L.W., (1998) Geographic Visualization: Designing Manipulable Maps for Exploring Temporally Varying Geo-referenced Statistics, *Proceedings, Information Visualization '98*. IEEE Computer Society, Raleigh-Durham, NC, Oct. 19-20, 1998, pp. 87-94.

Neter, J., Wasserman, W., and Kutner, M. (1983) *Applied Linear Regression Models*, Richard D. Irwin Inc., Homewood, IL.

Ryan, T.P. (1997) *Modern Regression Methods*, Wiley, New York

Seber, G.A.F and Wild, C.F. (1989) *Nonlinear Regression*, John Wiley and Sons, New York.