The Stata Journal (2001) **1**, Number 1, pp. 118

M statistic commands: interpoint distance distribution analysis

Pietro Tebaldi
Bocconi University
Milan, Italy
pietro.tebaldi@studbocconi.it

Marco Bonetti
Bocconi University
Milan, Italy
marco.bonetti@unibocconi.it

Marcello Pagano
Harvard School of Public Health
Boston, MA
pagano@hsph.harvard.edu

Abstract. We implement the commands *mstat* and *mtest* to perform inference based on the M statistic, a statistic that can be used to compare the interpoint distance distribution across groups of observations.

The analyses are based on the study of the interpoint distances between n points in a k-dimensional setting to produce a one-dimensional real-valued test statistic. The locations are distributed in a region of the plane, and when we consider all $\binom{n}{2}$ interpoint distances the dependencies among them are difficult to express analytically, but their distribution is informative and the M statistic can be built to summarize one aspect of this information.

The two commands can be used on a wide class of datasets to test the null hypothesis that two groups have the same (spatial) distribution. *mstat* and *mtest* return the exact M test statistic. Moreover, *mtest* executes a Monte Carlo type permutation test, returning the empirical *p-value* together with its confidence interval. This is the command to be used in most situations, since the convergence of M to its asymptotic *Chi-square* distribution is slow.

Both commands can be used to obtain graphical output of the empirical density function of the interpoint distance distributions in the two groups and/or the two-dimensional map of the n observations in the plane.

The descriptions of the commands are accompanied by examples of applications with real and simulated data. We run the test on the Alt and Vach gravesite dataset, rejecting the null hypothesis in contradiction to other published analyses. We also show how to adapt the techniques to discrete datasets with more than one unit in each location. Finally, we report an extensive application on breast cancer data in Massachusetts in which we show the compatibility of the M commands with Pisati's *spmap* package.

Keywords: mstat, mtest, M statistic, interpoint distance, Monte Carlo test, spmap

1 Introduction

The *mstat* and *mtest* commands are designed to implement inference based on the M statistic, a statistic that can be used to investigate the distribution of distances (interpoint distance distribution or IDD) between two observations and forms the building block of analyses that compare the IDD across groups of multivariate observations.

The M statistic is the result of a series of papers, starting with Bonetti and Pagano (2005) (2). The analyses are based on the study of the interpoint distances between n points in a k-dimensional setting to produce a one-dimensional, real-valued test statistic similar to Pearson's Chi-Squared goodness of fit statistic, that allows the comparison of such distributions across groups or to a null distribution. Thus the starting point, or data, for analysis is the $n \times n$ (symmetric) matrix of distances between the points, $\mathbf{D}_n$. This matrix can be replaced by any matrix containing a measure of similarity or dissimilarity between the points; the statistic does not exploit the triangle inequalities, for example, exhibited by a distance matrix, but for the sake of definiteness, we continue to focus on Euclidean distances on the plane. This is not necessary for the validity of the method: the data can consist of multivariate observations having high dimensionality, and the interpoint distance can take many different forms.

When we have the probability distribution of n points in a region of the plane, a complete description of the distribution of the pairwise distances between these points cannot be derived analytically, except for simple cases. The dependencies among these distances are very difficult to express analytically, but yet the distribution of the $\binom{n}{2}$ dependent distances is informative, and the M statistic is built to capitalize on this information.

Inference is based on the empirical cumulative density function (ECDF) of the $\binom{n}{2}$ dependent distances. In (2) the authors show that the ECDF of all pairwise distances evaluated at a finite number of values along the distance axis has an asymptotic multivariate normal distribution. This result is used to test whether the points in the sample follow the same (spatial) distribution as the underlying population. Manjourides(4) extends this result to the two samples case, that is to situations in which one wants to test wether or not two groups follow the same spatial distribution.

Inference based on M can be the building block for a wide class of empirical studies, from biosurveillance to economics, because of its power to detect situations where in some areas (clusters) the occurrence of an observed phenomenon is significantly higher or lower than in others. In particular, the method has been used in public health disease surveillance not only because of its power characteristics, but also because of its ability to analyze spatial data without the need to directly know the actual locations being investigated but referring instead to their interpoint distance only[1]. The M statistic has been shown to be effective at detecting exogenous clusters when compared with other statistics designed for the same purpose.

[1] This advantage is of course even more important when one analyzes (very) high dimensional data, whose distribution may be impossible to state.

In what follows we briefly summarize the theoretical results in (2) and (4). We then describe the new commands to be used to implement the M statistic method with Stata. The description is augmented with examples of applications in which we show how the method can be adapted to different datasets with very simple manipulations. Moreover, we show how the datasets compatible with the M statistic commands present all the elements required by Pisati's *spmap* package (7), thus enabling us to obtain informative graphical outputs to support the hypothesis testing procedures.

2 Background

The M statistic is built to describe the (spatial) distribution of n observations using the $\binom{n}{2}$ interpoint distances between these units. The distribution of these distances is not one-to-one onto, because the underlying spatial distribution is invariant to rotations and translations, but yet can be used to detect deviations from expected behavior. The M statistic measures differences in the distribution of the interpoint distances between cases and a null hypothesis distribution (one sample M), or between cases and a control group (two samples M).

The M statistic is constructed by considering all the $\binom{n}{2}$ interpoint distances between the observed cases. To check for goodness-of-fit, the data is discretized into a $k \times 1$ vector of the cumulative frequency distribution of the observed distances in k classes (bins, henceforth), and the vector is compared to the corresponding $k \times 1$ vector expected under the null hypothesis. For a vector $\mathbf{d} = (d_1, ..., d_k)$ of cutoff points along the distance axis, let $\mathbf{F}_n(\mathbf{d}) = (F_n(d_1), ..., F_n(d_k))$ be the vector of the empirical density functions in each bin:

$$F_n(d_\ell) = \frac{1}{\binom{n}{2}} \sum_{i<j} \mathbf{1}\left(d\left(X_i, X_j\right) \le d_\ell\right) \ , \ \ell = 1, ..., k \ ,$$

where $d(\cdot, \cdot) : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ is the (Euclidean) distance function and $\mathbf{1}(\cdot)$ the indicator function. The comparison between $\mathbf{F}_n(\mathbf{d})$ and the null hypothesis distribution, say $\mathbf{F}(\mathbf{d})$, is based on the quadratic form

$$M = \left(\mathbf{F}_n(\mathbf{d}) - \mathbf{F}(\mathbf{d})\right)^T \mathbf{S}^- \left(\mathbf{F}_n(\mathbf{d}) - \mathbf{F}(\mathbf{d})\right) \ ,$$

where $\mathbf{S}^-$ is a generalized inverse of the estimated variance-covariance matrix of $\mathbf{F}_n(\mathbf{d})$.

This statistic can be described as a Mahalanobis distance (thus the M) between the observed and the expected distribution of the distances discretized to the k bins. Under the null hypothesis if all the distances were independent M would be the usual Pearson's Chi Square test statistic. However, the $\binom{n}{2}$ distances are not independent. Bonetti and Pagano (2) prove that $\mathbf{F}_n(\mathbf{d})$ weakly converges to a multivariate normal distribution as $n \to \infty$. More specifically, they show that under the null hypothesis $\mathbf{F}_n(\mathbf{d}) = \mathbf{F}(\mathbf{d})$,

$$\sqrt{n}\left(\mathbf{F}_n(\mathbf{d}) - \mathbf{F}(\mathbf{d})\right) \xrightarrow{d} \mathcal{N}_k\left(\mathbf{0}, \mathbf{\Sigma}\right) \text{ as } n \to \infty,$$

where $\mathbf{\Sigma}$ is the variance-covariance matrix of $\mathbf{F}_n(\mathbf{d})$. In particular, for two cutoff points

d_a and d_b, the corresponding covariance term in $\mathbf{\Sigma}$ is

$$\sigma_{a,b} = 4E\left[\mathbf{1}\left\{d\left(X_1, X_2\right) \leq d_a, d\left(X_1, X_3\right) \leq d_b\right\}\right] - P\left(d\left(X_1, X_2\right) \leq d_a\right) P\left(d\left(X_1, X_3\right) \leq d_b\right).$$

The asymptotic distribution of M can thus be obtained: the quadratic form converges to a χ^2 with degrees of freedom equal to the rank of $\mathbf{\Sigma}\mathbf{\Sigma}^-$, that is the number of bins k^2. Empirical experience shows, however, that the convergence of M to the χ^2_k is slow. For this reason, it is often preferable to use empirical testing routines, such as Monte Carlo permutation tests.

The choice of the number of bins to be used in computing M has a direct impact on the power of the test, and we refer to (9) for a detailed analysis of this issue.

M can be used to detect a broad range of deviations from a given underlying spatial distribution and, as we mentioned above, the method has been adapted to a two samples setting to test for differences between the (spatial) distributions of two groups.

3 Two samples M statistic, mstat and mtest commands

In this section we describe the *mstat* and *mtest* commands implementing the M statistic method in the two samples setting to test

$$\begin{aligned} H_0 &: \textit{the two groups have the same (spatial) distribution ;} \\ H_1 &: \textit{the two groups do not have the same (spatial) distribution.} \end{aligned}$$

The two samples data are lists of couplets $\left(\left(X_1, G_1\right), \left(X_2, G_2\right), ..., \left(X_n, G_n\right)\right)$ where X_i is the location of observation i and G_i is a *group indicator* variable taking the two values,

$$G_i = \begin{cases} 1, & \text{if subject } i \text{ is in Group 1} \\ 0, & \text{if subject } i \text{ is in Group 2.} \end{cases}$$

Let n_1 and n_2 be the number of subjects in Group 1 and 2, respectively, so that $n = n_1 + n_2$. Let also $\widehat{F}_{n_j}(d)$ be the ECDF computed using the subjects in Group j only ($j = 1, 2$), with the corresponding vector notation $\widehat{\mathbf{F}}_{n_j}(\mathbf{d})$ following intuitively.

Both commands, *mstat* and *mtest*, compute $\widetilde{M}$ for the two samples case:

$$\widetilde{M} = \left(\widehat{\mathbf{F}}_{n_1}(\mathbf{d}) - \widehat{\mathbf{F}}_{n_2}(\mathbf{d})\right)^T \ \mathbf{S}^- \ \left(\widehat{\mathbf{F}}_{n_1}(\mathbf{d}) - \widehat{\mathbf{F}}_{n_2}(\mathbf{d})\right) \qquad (1)$$

where $\mathbf{d}$ is a vector of cutoff points $\mathbf{d} = (d_1, d_2, ..., d_k)$ at which the ECDFs $\widehat{\mathbf{F}}_{n_1}(\cdot)$ and $\widehat{\mathbf{F}}_{n_2}(\cdot)$ are evaluated, and $\mathbf{S}^-$ is the Moore-Penrose generalized inverse of the estimated variance-covariance matrix $\mathbf{\Sigma}$ of the vector $\left(\widehat{\mathbf{F}}_{n_1}(\mathbf{d}) - \widehat{\mathbf{F}}_{n_2}(\mathbf{d})\right)$.

[2] Note that the midpoint algorithm used to define $d_1, ..., d_k$ (see next Section) is such that $\widehat{F}(d_k) < 1$, so that we are actually defining $(k+1)$ bins with the last being $[d_k, \infty)$.

The cutoffs $d_1, d_2, ..., d_k$ at which we evaluate the differences $\widehat{F}_{n_1}(d_\ell) - \widehat{F}_{n_2}(d_\ell)$, $\ell = 1, ..., k$ are the midpoints of k *equiprobable* bins: from the pooled sample we partition the range of $d(X_i, X_j)$ in k intervals $\mathbf{I}_m = [x_{1,m}, x_{2,m})$ so to have approximately

$$\frac{1}{\binom{n}{2}} \sum_{i<j} \mathbf{1}\left(d\left(X_i, X_j\right) \in \mathbf{I}_m\right) = \frac{1}{k} \ , \ \forall m = 1, ..., k \ ; \tag{2}$$

the elements of $\mathbf{d}$ are then $d_\ell = \frac{1}{2}\left(x_{2,\ell} - x_{1,\ell}\right)$, $\ell = 1, ..., k$.

Both algorithms start by computing the pooled sample distance matrix $\mathbf{D}_n$ with general entry $\mathbf{D}_n[i,j] = d(X_i, X_j)$ and generate the vector $\mathbf{d}$; these two tasks are executed by the subcommands *eucldist* and *dbins*, respectively.

The ECDFs for the two groups are then $\widehat{\mathbf{F}}_{n_1}(\mathbf{d}) = \left(\widehat{F}_{n_1}(d_1), \widehat{F}_{n_1}(d_2), ..., \widehat{F}_{n_1}(d_k)\right)$ and $\widehat{\mathbf{F}}_{n_2}(\mathbf{d}) = \left(\widehat{F}_{n_2}(d_1), \widehat{F}_{n_2}(d_2), ..., \widehat{F}_{n_2}(d_k)\right)$ with

$$\widehat{F}_{n_1}(d_\ell) = \frac{1}{\binom{n_1}{2}} \sum_{i<j} \mathbf{1}\left(d\left(X_i, X_j\right) \leq d_\ell\right) G_i G_j \ , \ \ell = 1, ..., k$$

and

$$\widehat{F}_{n_2}(d_\ell) = \frac{1}{\binom{n_2}{2}} \sum_{i<j} \mathbf{1}\left(d\left(X_i, X_j\right) \leq d_\ell\right)(1 - G_i)(1 - G_j) \ , \ \ell = 1, ..., k.$$

The subcommand *Fhat* evaluates the ECDF of the interpoint distances for a user-given vector of cutoff points $\mathbf{d}$, and the subcommand _*diff* returns the $k \times 1$ vector $\left(\widehat{\mathbf{F}}_{n_1}(\mathbf{d}) - \widehat{\mathbf{F}}_{n_2}(\mathbf{d})\right)$.

The variance-covariance matrix of $\left(\widehat{\mathbf{F}}_{n_1}(\mathbf{d}) - \widehat{\mathbf{F}}_{n_2}(\mathbf{d})\right)$, $\mathbf{\Sigma}$, is estimated under the null hypothesis: the general entry derived in (4) is

$$\widehat{Cov}\left(\widehat{F}_{n_1}(d_a) - \widehat{F}_{n_2}(d_a), \widehat{F}_{n_1}(d_b) - \widehat{F}_{n_2}(d_b) | H_0\right) =$$

$$\widehat{\sigma}_{a,b} = \left(\frac{n}{n_1 n_2}\right) \frac{4}{\binom{n}{3}} \sum_{i<j,k} \mathbf{1}\left(d\left(X_i, X_j\right) \leq d_a, d\left(X_i, X_k\right) \leq d_b | H_0\right).$$

The subcommand *Smat* returns the $k \times k$ matrix $\mathbf{S}$, and it is based on the following algorithm:

1. For each cutoff value d_ℓ, generate an $n \times n$ indicator matrix $\mathcal{I}_\ell$, whose general entry is $\mathcal{I}_\ell[i,j] = \mathbf{1}\left(d\left(X_i, X_j\right) \leq d_\ell\right)$[3].

[3] This requires Mata to generate k matrices, each of which results from executing n^2 inequalities. This long loop is the main reason why the user can experience a long execution time when n is large.

2. For each pair (a, b) , $a, b = 1, ..., k$, take the matrix product $\mathcal{I}_a \cdot \mathcal{I}_b$ and sum all the elements of the resulting $n \times n$ matrix.
3. Divide the resulting value by $\lambda = 2\binom{n}{2} + n(n-1)(n-2)$, that is the same as we were dividing by $\binom{n}{3}$ while adjusting for all the repeated values that should not be considered in the summation: calling $\mathbf{1}_n$ the $n \times 1$ unity vector we have
$$\frac{\mathbf{1}_n' \cdot (\mathcal{I}_a \cdot \mathcal{I}_b) \cdot \mathbf{1}_n}{\lambda} = \frac{\sum_{i<j,k} \mathbf{1}\left(d\left(X_i, X_j\right) \le d_a, d\left(X_i, X_k\right) \le d_b\right)}{\binom{n}{3}}.$$
4. Compute the $k \times k$ general entries as $\widehat{\sigma}_{a,b} = 4\left(\frac{n}{n_1 n_2}\right) \frac{1}{\lambda} \mathbf{1}_n' \cdot (\mathcal{I}_a \cdot \mathcal{I}_b) \cdot \mathbf{1}_n$.

Once $\mathbf{S}$ has been obtained as just described, $\mathbf{S}^-$ is the corresponding Moore-Penrose generalized inverse (Mata function *pinv()*).

Command *mstat* computes the value of the test statistic $\widetilde{M}$, and, if required (option *chi2*), returns the *p-value* of the asymptotic Chi-square test[4].

mtest executes *mstat*. It runs then a Monte Carlo-type test to come up with an empirical *p-value*: the values of the binary variable G are randomly permuted a user-defined number of times NP; each time the empirical M statistic is computed, generating a vector $(M_1, M_2, ..., M_{NP})$. Under the null hypothesis $\widetilde{M}$ should not be significantly larger than these values: the empirical *p-value* is computed as

$$p\text{-}value = \frac{1}{NP} \sum_{s=1}^{NP} \mathbf{1}\left(M_s \ge \widetilde{M}\right) \ .$$

3.1 Example: Alt and Vach data

We show an application of *mtest* taken from Section 1.5 of (4). The author uses the Alt and Vach data (reduced dataset from (8)) to examine the spatial distribution of corpses found in a medieval grave site in Neresheim, Baden-Wurttemberg, Germany. The problem of interest was a kinship analysis to determine if members of the same family are buried close together. The discrimination between the two groups is based on a dental defect that was found in 30 out of 143 corpses in the grave site area. As pointed out in (4), spatial methods such as the Wilcoxon rank-sum test, the Kelsall and Diggle test, or the K-function test (SaTScan), provide little or weak evidence against the null hypothesis of no difference between the two spatial distributions.

[4] Because the convergence of $\left(\widehat{\mathbf{F}}_{n_1}(\mathbf{d}) - \widehat{\mathbf{F}}_{n_2}(\mathbf{d})\right)$ to a k-variate $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ is *slow*, the asymptotic Chi-square test is to be used only with n significantly large, and in that case the command *mtest* will require a long execution time.

When one runs *mtest* on the data, however, a *p-value* smaller than 0.05 is obtained (1000 permutations). The heuristic analysis of the kernel densities of the interpoint distances and the scatter plot of the two groups helps our understanding.

Below is the output from running the command. The observed M is 79.64, consistent with the result obtained in (4). The Monte Carlo permutation test with 1000 iterations returns a *p-value* of 0.023, and the 95% "exact" binomial confidence interval for this *p-value* is $[0.015, 0.034]$.

As shown in Figure 1 a peculiar characteristic of this data is the shape of the area of interest. In Figure 2 we report the Kernel densities of the interpoint distances within the two groups.

```
-----------------------------------------------------------------
M statistic
Monte Carlo permutation results
H0: The two groups have the same spatial distribution
Number of bins = 20
Number of permutations = 1000
-----------------------------------------------------------------
     M(obs) |      c       n  p=c/n    SE(p) [95% Conf. Interval]
------------+----------------------------------------------------
   79.63794 |     23    1000  0.0230  0.0047  .0146346   .0343123
-----------------------------------------------------------------
note: c = #{M>=M(obs)}
note: exact binomial confindece interval with respect to p=c/n
```

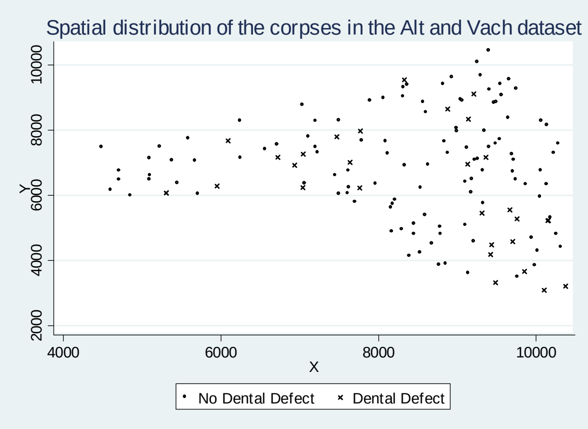

Figure 1: Scatter plot obtained by calling option *scatter* and the related graphic options.

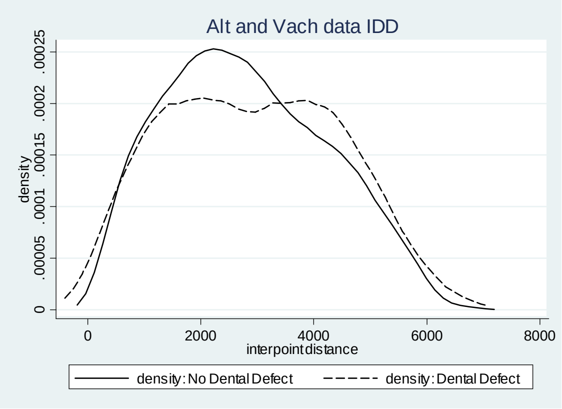

Figure 2: Kernel densities obtained by calling option *density* and the related graphic options.

3.2 Dealing with discrete datasets

In many applications spatial data are aggregated in a discrete number of locations, say $l_1, l_2, ..., l_m$, (typically ZIP codes, census tracts, etc.). The extended analysis of the discrete case in Reference (2) shows that M can be applied to this framework.

To implement the M statistic method in this discrete setting with Stata, the data need to be adapted to the standard requirements for *mstat* or *mtest*. A typical discrete dataset has the form

$$((l_1, n_{1,1}, n_{2,1}), ..., (l_m, n_{1,m}, n_{2,m}))$$

where $l_s = (x_s, y_s)$ are the coordinates of location s, and $n_{1,s}, n_{2,s}$ represent the numbers of individuals in location s belonging to groups 1 and 2, respectively. This specification is indeed the most general, as it includes the case of all observations having different coordinates.

To execute the M commands on these data we need to have a dataset of the form introduced above:

$$((X_1, G_1), ..., (X_n, G_n)).$$

We propose two procedures: one for the a small number of individuals, and one for a large number of individuals.

Small to moderate number of individuals

With discrete data, the total number of individuals is $N = \sum_{s=1}^{m} n_s$, where $n_s = n_{1,s} + n_{2,s}$. To execute the M commands we need to *expand* the dataset so to have N

observations, each of them including the corresponding coordinates and group dummy variable.

In the following example we illustrate the procedure on a simulated dataset with the two groups aggregated in 12 locations:

```
. list _all

     +-------------------------------+
     |        X          Y   n2   n1 |
     |-------------------------------|
  1. | .3734917    .366027   10    7 |
  2. | .2792006   .8637221    4   10 |
  3. | .8315064   .5910658    8    3 |
  4. | .9711422   .6305301   13   11 |
  5. | .7767971    .175099   18    9 |
     |-------------------------------|
  6. |  .643114   .1090542    3    7 |
  7. | .3833295   .6420991   16   10 |
  8. | .0057233   .5137199    5    7 |
  9. | .8772233   .8745837   20    3 |
 10. | .6526399        .25    8    8 |
     |-------------------------------|
 11. | .2033027   .4896181   12    7 |
 12. | .6363281   .8478178    2    8 |
     +-------------------------------+

. */ generate two observations for each location and the Group variable
. expand 2 , gen(G)
(12 observations created)
. */ expand Group 1 observations
. expand n1 if G==1
(78 observations created)
. */ expand Group 2 observations
. expand n2 if G==0
(107 observations created)
. */ execute mstat
. mstat , x(X) y(Y) g(G) chi2
----------------------------------------------------------------------
M statistic
Number of bins = 20
----------------------------------------------------------------------
M = 25.108731                      Chi2(20) = .19730295
----------------------------------------------------------------------
```

Large Number of Individuals

Suppose that the number of individuals distributed in the m locations is very large, and that we want to use the M commands *mstat* or *mtest*. We propose the following algorithm:

1. Consider the variables $n_{1,\cdot}$ and $n_{2,\cdot}$ in relative terms, i.e. construct for each location the proportions

   $$p_{1,s} = \frac{n_{1,s}}{N_1} \ , \ p_{2,s} = \frac{n_{2,s}}{N_2} \ , \ s = 1, ..., m \ ;$$

   where $N_1 = \sum_{s=1}^{m} n_{1,s}$ and $N_2 = \sum_{s=1}^{m} n_{2,s}$.

2. Transform these proportions in integer terms, i.e. multiply by $k \geq 100$ (for instance 500 or 1000, but much smaller than N) and round them to the closest integer. The larger the factor by which the proportions $p_{1,s}$ and $p_{2,s}$ are multiplied, the less information is lost due to rounding.

3. Expand the data, generating for each location l_s (for each group) as many observations as the integer values derived above.

4. Execute *mstat* or *mtest* on these data.

The procedure transforms the discrete dataset into a dataset compatible with the M command composed of $2k$ observations, k in each group, preserving the proportions $p_{1,s}$ and $p_{2,s}$ across the different locations. Because the M statistic is based on the ECDFs of the interpoint distances and the relative frequencies in each bin are invariant to these manipulations, so is the value of M. Note that the test is based on equal sized groups, and with $2k$ significantly smaller than N the power will be lower. Indeed, for very large N_1 and N_2 one would realistically almost always reject the null, as the null is essentially never true in real life settings. Hence it is of interest to observe whether for feasible values of k one does indeed "already" reject the null.

To implement the algorithm the user can refer to the following example in which we created data with sharp differences across the two groups:

```
. list _all
     +-----------------------------------+
     |        X           Y     n2    n1 |
     |-----------------------------------|
  1. | .3734917     .366027    320    22 |
  2. | .2792006    .8637221   1067   250 |
  3. | .8315064    .5910658    150    27 |
  4. | .9711422    .6305301    870    26 |
  5. | .7767971     .175099   1050   100 |
     |-----------------------------------|
  6. |  .643114   .1090542    900   340 |
  7. | .3833295   .6420991    810   250 |
  8. | .0057233   .5137199    630   120 |
  9. | .8772233   .8745837   1200    31 |
 10. | .6526399        .25   1800   400 |
     |---------------------------------|
 11. | .2033027   .4896181    240    20 |
 12. | .6363281   .8478178    700   130 |
     +---------------------------------+
. total n2 n1
Total estimation                    Number of obs    =      12

--------------------------------------------------------------
             |      Total   Std. Err.     [95% Conf. Interval]
-------------+------------------------------------------------
          n2 |       9737   1588.352       6241.06    13232.94
          n1 |       1716   466.8214      688.5331    2743.467
--------------------------------------------------------------
. */ generate proportions and round them (k=100)
. gen ps2 = round(n2/9737*100)
. gen ps1 = round(n1/1716*100)
. */ generate two observations for each location and the Group variable
. expand 2 , gen(G)
(12 observations created)
. */ expand Group 1 observations
. expand ps1 if G==1
(90 observations created)
. */ expand Group 2 observations
. expand ps2 if G==0
(86 observations created)
. */ execute mtest
. mtest , x(X) y(Y) g(G) den
-----------------------------------------------------------------
M statistic
Monte Carlo permutation results
H0: The two groups have the same spatial distribution
Number of bins = 20
Number of permutations = 100
-----------------------------------------------------------------
     M(obs) |      c       n   p=c/n   SE(p) [95% Conf. Interval]
------------+----------------------------------------------------
   22.00803 |      2     100  0.0200  0.0140  .0024313   .0703839
-----------------------------------------------------------------
note: c = #{M>=M(obs)}
note: exact binomial confindece interval with respect to p=c/n
```

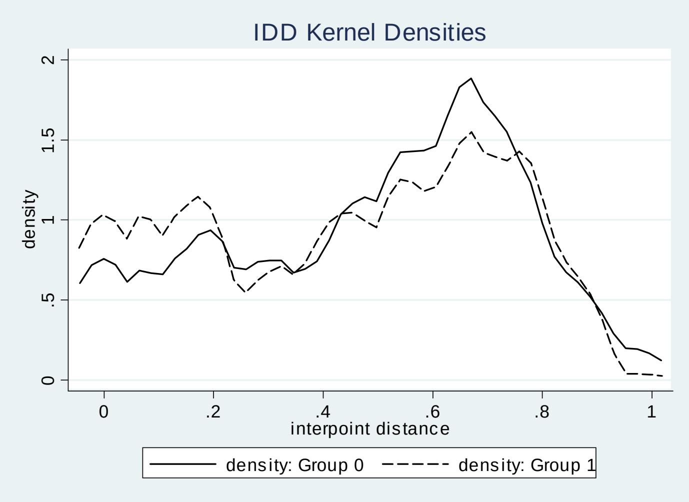

Figure 5: Kernel densities for Group 0 and 1 in the large sample setting with discrete data.

4 Application: Breast Cancer Data in Massachusetts

In this section we report on an application of *mtest* in which we also show the compatibility of the data format required by our new commands with the format required by Pisati's *spmap* package. We use two datasets:

a) A dataset resulting from the composition of Breast Cancer Data extracted from (5) with Census Tracts coordinates for the State of Massachusetts;

b) Scott Merryman's US county map coordinates for Pisati's package.

a) In search of an easily available dataset, we consider 348 locations in the State of Massachusetts for which detailed cancer data are available. For each location the Massachusetts Cancer Registry reports the counts of the observed and the expected cases, the latter being "*a calculated number based on the city/town's population distribution (by sex and among eighteen age groups) for the time period 2002-2006, and the corresponding statewide average annual age-specific incidence rates*"(5). We focus here on breast cancer among females only, a cancer site (and gender) suspected of being clustered in specific areas of the State (see for instance http://www.mbcc.org/ or http://www.womensenews.org/story/health/020712/ researchers-probe-cape-cods-breast-cancer-rate).

To simplify the analysis, we consider the variable

$$rel_s = \frac{\text{observed}_s \ - \ \text{expected}_s}{\sqrt{\text{expected}_s}} \ , \ s = 1, ..., 348,$$

a standardized difference between observed and expected numbers of cases.

The variable *rel* is approximately normally distributed with zero mean. This is confirmed by our inspection of the data if we ignore the geographical positioning. The null hypothesis implies that *there are no clusters in the spatial distribution of breast cancer in the female population in Massachusetts.* Thus there should be no difference between the distribution of locations with *rel>0* and locations with *rel≤0.* This sign test is equivalent to testing that the groups have the same spatial distribution, with the group dummy variable being

$$G_s = \mathbf{1}\,(rel_s > 0) \ , \ s = 1, ..., 348.$$

Once G is generated, and matching each location with the corresponding census tract coordinates, we have a dataset featuring the structure required by the M commands.

b) We can also use the dataset together with Pisati's *spmap* package (Scott Merryman's polygons data(6)) to map the distribution of G. We report here the do-file to execute the test on the data. Instead of calling option *scatter*, we use *spmap* to superimpose variable G on the map of Massachusetts together with the locations of the 31 EPA Superfund sites in the State[5].

```
-----------------------------------------------------------------
M statistic
Monte Carlo permutation results
H0: The two groups have the same spatial distribution
Number of bins = 20
Number of permutations = 1000
-----------------------------------------------------------------
     M(obs) |      c       n   p=c/n   SE(p) [95% Conf. Interval]
------------+----------------------------------------------------
   38.56281 |      2    1000  0.0020  0.0014  .0002423   .0072058
-----------------------------------------------------------------
note: c = #{M>=M(obs)}
note: exact binomial confindece interval with respect to p=c/n
```

[5] This third dataset is created by associating each of the sites with the corresponding coordinates in the Census Tracts Dataset.

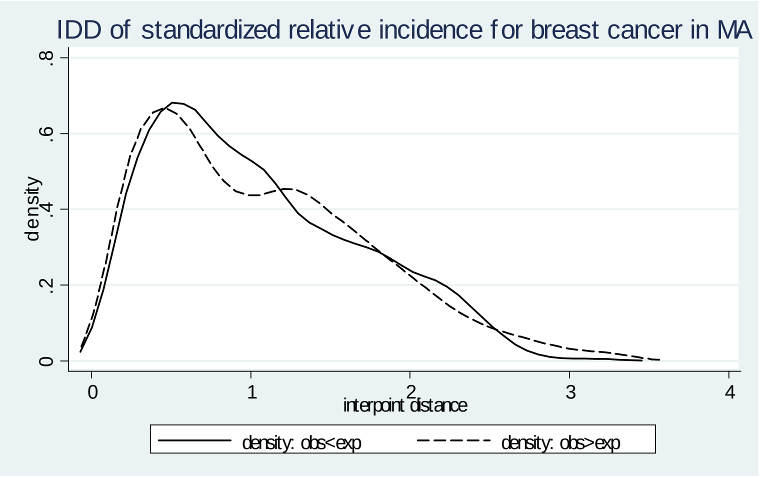

Figure 6: Kernel densities. Units in Group 1 are locations with observed cases exceeding the expected.

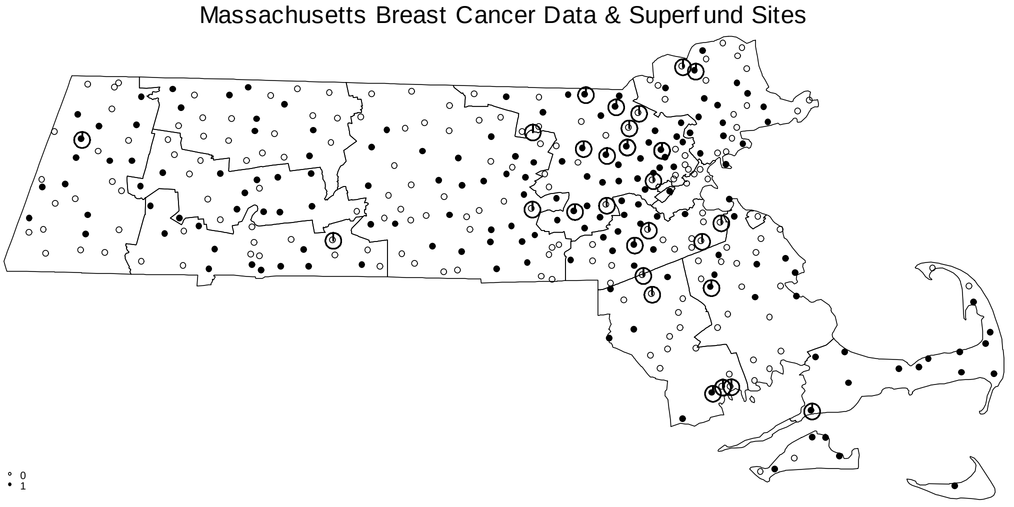

Figure 7: Map of Massachusetts with the black (empty) dots corresponding to locations with observed cases (not) exceeding the expected. The circles around some of the locations indicate the presence of an EPA superfund site.

Our results based on M (reported *p-value*$= 0.002$ with confidence interval $[0.0002, 0.007]$) indicate that there are indeed areas of Massachusetts in which the incidence of breast cancer is consistently and significantly above the expected number.

5 mstat and mtest: syntax, options and returned results

The syntax for the two commands is the following:

`mstat , x(`*varname*`) y(`*varname*`) g(`*varname*`) [`*options graphic options*`]`

`mtest , x(`*varname*`) y(`*varname*`) g(`*varname*`) [`*options graphic options*`]`

where `x(`*varname*`) y(`*varname*`) g(`*varname*`)` are three required options indicating the x-coordinates, the y-coordinates and the group dummy variable, respectively. The other command-specific options are

`Options for mstat`

`bins(#)` choose the number of bins to be used

`chi2` returns the upper tail p-value of the asymptotic chi2 distribution

`scatter` plot the spatial distribution of the two groups

`density` plot the Kernel density for the interpoint distances

`Options for mtest`

`bins(#)` choose the number of bins to be used

`iter(#)` choose the number of Monte Carlo permutations in the test

`level(#)` choose the level for the p-value confidence interval to be reported

`scatter` plot the spatial distribution of the two groups

`density` plot the Kernel density for the interpoint distances

Notice that option `bins(#)` selects, in both cases, the number of bins. This choice affects the power of the test, and the way in which it does that can change depending on the dataset being tested. The default value is `bins(20)`. For a detailed analysis of the implications of changing the number of bins we refer to Forsberg et al. (9).

Option `iter(#)`, instead, does not affect the power of the test but, by setting the number of Monte Carlo permutations to be executed, it determines the accuracy of the empirical p-value. The default value is iter(100).

Option `level(#)` set the confidence level at which the "exact" binomial confidence interval of the Monte Carlo p-value is constructed. The default is level(95).

For both commands, when the graphic output is generated by calling option `scatter` or `density`, the following graphic options are available:

`Options when scatter is specified`

`scolor0(colorstyle)` set the color for the marker of group 0.

`scolor1(colorstyle)` set the color for the marker of group 1.

`smarker0(marker symbol)` set the symbol for the marker of group 0.

`smarker1(marker symbol)` set the symbol for the marker of group 1.

`ssize0(marker size)` set the size for the marker of group 0.

`ssize1(marker size)` set the size for the marker of group 1.

`slabel0(string)` input the label for group 0 in the legend, default: "Group 0".

`slabel1(string)` input the label for group 1 in the legend, default: "Group 1".

`stitle(string)` specifies the title for the scatter, default: "Spatial Distribution of the two groups".

`sytitle(string)` specifies the title for the y axis, default is the name of the variable in option `y(y-coord)`.

`sxtitle(string)` specifies the title for the x axis, default is the name of the variable in option `x(x-coord)`.

`Options when density is specified`

`dcolor0(colorstyle)` set the color for the line of the density of group 0.

`dcolor1(colorstyle)` set the color for the line of the density of group 1.

`dpattern0(line pattern style)` set the pattern style for the line of the density of group 0.

`dpattern1(line pattern style)` set the pattern style for the line of the density of group 1.

`dwidth0(line width style)` set the width for the line of the density of group 0.

`dwidth1(line width style)` set the width for the line of the density of group 1.

`dlabel0(string)` input the label for group 0 in the legend, default: "Group 0".

`dlabel1(string)` input the label for group 1 in the legend, default: "Group 1".

`dtitle(string)` specifies the title for the Kernel density, default: "IDD Kernel Densities".

`Saved results`

`mstat` saves the following in `r()`:

`Scalars`	
`r(M)`	observed M statistic
`r(p)`	chi-squared p-value (if option chi2 is specified)
`Matrices`	
`r(difF)`	difference between the ECDFs in the two groups
`r(Sinv)`	generalized inverse of the covariance matrix of r(difF)
`r(d)`	cutoffs of the equiprobable bins

`mtest` saves the following in `r()`:

`Scalars`	
`r(N)`	sample size
`Matrices`	
`r(M)`	observed M statistic
`r(c)`	count when M>=M(obs) is true
`r(p)`	observed empirical p-value
`r(se)`	standard error of empirical p-value
`r(ci)`	exact binomial confidence interval of observed p-value
`r(reps)`	number of nonmissing results
`r(d)`	cutoffs of the equiprobable bins
`r(Sinv)`	generalized inverse of the covariance matrix

6 Conclusions

mstat and *mtest* allow Stata users to use powerful tests for detecting differences between the spatial distribution of two groups. So far the M test has been used (in its one sample version) for excluding the presence of clusters in the population, in particular in epidemiological studies. The one sample version of the test requires the user to have knowledge of the null distribution, i.e. the interpoint distance distribution that is compared to the observed one, or at least the density in each bin. Since often this is not available, except when we work with census data (e.g. (2)), in several applications the null distribution is either simulated or estimated. In the latter case, that is when we have a dataset being used to estimate the underlying null distribution, the one sample M test is almost equivalent to a two samples M test where the groups correspond to the two datasets, the only difference being that in the two samples case when computing the matrix **S** we take into consideration the variability in both groups. The loop for the construction of this matrix **S** is the core of both commands, with the rest of the algorithm being based on a sequence of existing Stata commands.

The two commands presented here deal with datasets in the two-dimensional Euclidean space only. Since the M statistics method does not depend on this fact, the method works with any kind of dissimilarity measure. An extension of the commands could allow the user to test differences in the interpoint distance distribution in k-dimensional Euclidean spaces and, more generally, with generic dissimilarity measures. Alternatively, one could also develop a shorter version of the command so as to have the user inputting directly the dissimilarity matrices **D** (one for each group) thus allowing for the greatest level of generality. Clearly, all these extensions to higher dimensional settings would prevent the possibility of a simple graphic output to support the numerical results, as it is the case for *mstat* and *mtest*.

The M statistics method may prove valuable in several fields, whenever detecting situations in which the distribution of certain phenomena in space is not trivial, yet relevant. Thanks to the latest advancements in bioinformatics, for instance, statistical studies in genetics can be based on the difference between the distribution of a dissimilarity measure between genetic sequences in two groups. Sociology, demography and economics are other fields in which detecting differences in the (spatial) distribution of different groups of individuals is certainly relevant.

7 Acknowledgments

The authors would like to thank Justin Manjourides, Al Ozonoff and Sergio Venturini for their helpful suggestions on the *Smat* auxiliary command and on the graphic options of the commands. Our research was supported in part by a grant from the NIH, P01 CA 134294.

8 References

[3] Bonetti, M., Forsberg, L., Ozonoff, A. and Pagano, M., *The distribution of interpoint distances.* Mathematical Modeling Applications in Homeland Security. HT Banks and C Castillo-Chavez, Eds. ; 2003:87-106.

[2] Bonetti, M. and Pagano, M., *The interpoint distance distribution as a descriptor of point patterns, with an application to spatial disease clustering.* Stat Med 2005; 24(5):753-773.

[3] Forsberg, L., Bonetti, M., Jeffery, C., Ozonoff, A. and Pagano, M., *Distance-Based Methods for Spatial and Spatio-Temporal Surveillance.* Spatial and Syndromic Surveillance for Public Health (ch.8). John Wiley & Sons, Ltd; 2005.

[4] Manjourides, J., *Distance based methods for space time modeling of the health of populations.* Harvard University, Cambridge, MA, 2009.

[5] Massachusetts Cancer Registry, City and Town Series 2002-2006, http://www.mass.gov/dph/mcr

[6] Merryman, S., *USMAPS2: Stata module to provide US county map coordinates for tmap*, Statistical Software Components, Boston College Department of Economics; 2005.

[7] Pisati M., *Simple thematic mapping.* The Stata Journal 2004; 4: 361-378.

[8] Waller, L. and Gotway, C., *Applied Spatial Statistics for Public Health Data.* Wiley-IEEE, 2004.

[9] White, L.F., Bonetti, M. and Pagano, M., *The choice of the number of bins for the M statistic.* Computational Statistics & Data Analysis 2009; 53(10): 3640-3649.