The Stata Journal (2011)
11, Number 2, pp. 271–289

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. 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, which returns the empirical p-value together with its confidence interval. This is the command to use in most situations, because the convergence of M to its asymptotic chi-squared 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 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 grave site dataset (Manjourides and Pagano, forthcoming, *Statistics in Medicine*) and reject 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 the application, we show the compatibility of the M commands with Pisati's `spmap` package.

Keywords: st0228, 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 [IDD]) between two observations and can form building blocks of analyses that compare the IDDs across groups of multivariate observations.

The M statistic is the result of a series of papers that started with Bonetti and Pagano (2005). 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 with a null distribution. Thus the starting point, or data, for analysis is the $\mathbf{n} \times \mathbf{n}$ 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 focus 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 in simple cases. The dependencies among these distances are very difficult to express analytically, but 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 Bonetti and Pagano (2005), 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 and Pagano (forthcoming) extends this result to two-sample cases, that is, to situations in which one wants to test whether 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 only to their interpoint distances.[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 Bonetti and Pagano (2005) and Manjourides and Pagano (forthcoming). 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 (2004) `spmap` package, 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, yet it 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-sample 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 are 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 with the corresponding $k \times 1$ vector expected under the null hypothesis. For a vector $\mathbf{d} = (d_1, \ldots, d_k)$ of cutoff points along the distance axis, let $\mathbf{F}_n(\mathbf{d}) = \{F_n(d_1), \ldots, F_n(d_k)\}$ be the vector of the empirical density functions in each bin:

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

where $d(\cdot,\cdot) : R^2 \times R^2 \to R$ is the (Euclidean) distance function and $\mathbf{1}(\cdot)$ is 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

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

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-squared test statistic. However, the $\binom{n}{2}$ distances are not independent. Bonetti and Pagano (2005) 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 $\mathbf{F}_n(\mathbf{d}) = \mathbf{F}(\mathbf{d})$,

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

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

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

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 used in computing M has a direct impact on the power of the test, and we refer to White, Bonetti, and Pagano (2009) 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-sample setting to test for differences between the (spatial) distributions of two groups.

3 Two-sample M statistic—mstat and mtest commands

In this section, we describe the `mstat` and `mtest` commands implementing the M statistic method in the two-sample setting to test

H_0: The two groups have the same (spatial) distribution.
H_a: The two groups do not have the same (spatial) distribution.

The two-sample data are lists of couplets $\{(X_1,G_1),(X_2,G_2),\ldots,(X_n,G_n)\}$ where X_i is the location of observation i, and G_i is a group-indicator variable that may take 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 groups 1 and 2, respectively, so that $n = n_1 + n_2$. Also let $\widehat{F}_{n_j}(d)$ be the ECDF computed using only the subjects in group j ($j = 1, 2$), 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-sample case:

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

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

where $\mathbf{d}$ is a vector of cutoff points $\mathbf{d} = (d_1, d_2, \ldots, d_k)$ 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 $\boldsymbol{\Sigma}$ of the vector $\left\{\widehat{\mathbf{F}}_{n_1}(\mathbf{d}) - \widehat{\mathbf{F}}_{n_2}(\mathbf{d})\right\}$.

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

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

The elements of $\mathbf{d}$ are then $d_\ell = 1/2\,(x_{2,\ell} - x_{1,\ell})$, $\ell = 1, \ldots, k$.

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 by generating the vector $\mathbf{d}$; these two tasks are executed by the subcommands `eucldist` and `dbins`, respectively.

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

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

and

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

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 $\left\{\widehat{\mathbf{F}}_{n_1}(\mathbf{d}) - \widehat{\mathbf{F}}_{n_2}(\mathbf{d})\right\}$, $\boldsymbol{\Sigma}$, is estimated under the null hypothesis: the general entry derived in Manjourides and Pagano (forthcoming) is

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

The `smat` auxiliary command returns the $\mathbf{k} \times \mathbf{k}$ matrix $\mathbf{S}$, and it is based on the following algorithm:

1. For each cutoff value d_ℓ, generate an $\mathbf{n} \times \mathbf{n}$ indicator matrix $\mathcal{I}_\ell$, whose general entry is $\mathcal{I}_\ell(i,j) = \mathbf{1}\{d(X_i, X_j) \leq d_\ell\}$.[3]
2. For each pair (a,b), $a,b = 1,\ldots,k$, take the matrix product $\mathcal{I}_a \cdot \mathcal{I}_b$ and sum all the elements of the resulting $\mathbf{n} \times \mathbf{n}$ matrix.
3. Divide the resulting value by $\lambda = 2\binom{n}{2} + n(n-1)(n-2)$. That is the same as if 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
$$\frac{\mathbf{1}_n' \cdot (\mathcal{I}_a \cdot \mathcal{I}_b) \cdot \mathbf{1}_n}{\lambda} = \frac{\sum_{i<j,k} \mathbf{1}\{d(X_i, X_j) \leq d_a, d(X_i, X_k) \leq d_b\}}{\binom{n}{3}}$$
4. Compute the $k \times k$ general entries as $\widehat{\sigma}_{a,b} = 4(n/n_1 n_2)(1/\lambda)\,\mathbf{1}_n' \cdot (\mathcal{I}_a \cdot \mathcal{I}_b) \cdot \mathbf{1}_n$.

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

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

`mtest` executes `mstat`. It then runs 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 and a vector $(M_1, M_2, \ldots, M_{\text{NP}})$ is generated. Under the null hypothesis, $\widetilde{M}$ should not be significantly larger than these values. The empirical p-value is computed as
$$p\text{-value} = \frac{1}{\text{NP}} \sum_{s=1}^{\text{NP}} \mathbf{1}\left(M_s \geq \widetilde{M}\right)$$

3.1 Example: Alt and Vach data

We show an application of `mtest` taken from section 1.5 of Manjourides and Pagano (forthcoming). The author uses the Alt and Vach data (reduced dataset from Waller and Gotway [2004]) to examine the spatial distribution of corpses found in a medieval grave site in Neresheim, Baden-Württemberg, Germany. The problem of interest is a kinship analysis to determine if members of the same family are buried close together.

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.
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 $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ is slow, the asymptotic chi-squared test is to be used only with an n that is significantly large, and in that case, the `mtest` command will require a long execution time.

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 Manjourides and Pagano (forthcoming), 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.

When one runs `mtest` on the data, however, a p-value smaller than 0.05 is obtained from 1,000 permutations. The heuristic analysis of the kernel densities of the interpoint distances and the scatterplot of the two groups helps our understanding.

Below is the output from running the command. The observed M is 79.64, which is consistent with the result obtained in Manjourides and Pagano (forthcoming). The Monte Carlo permutation test with 1,000 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 these 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 confidence interval with respect to p=c/n
```

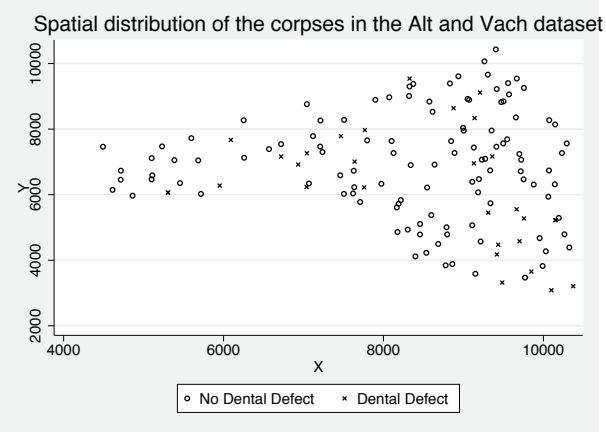

Figure 1. Scatterplot obtained by calling the `scatter` option and the related graphic options.

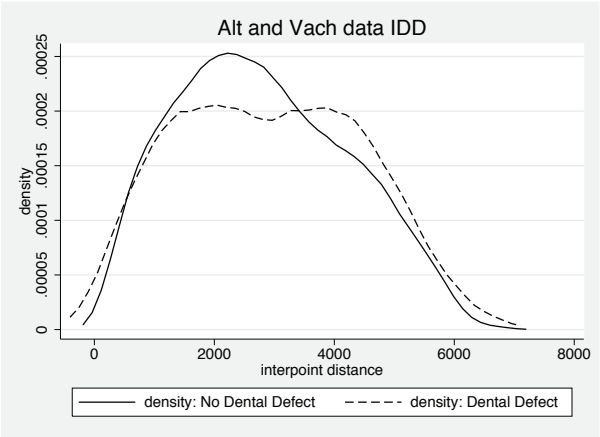

Figure 2. Kernel densities obtained by calling the `density` option and the related graphic options.

3.2 Dealing with discrete datasets

In many applications, spatial data are aggregated in a discrete number of locations, say, $l_1, l_2, \ldots, l_m$ (typically ZIP codes, census tracts, etc.). The extended analysis of the discrete case in Bonetti and Pagano (2005) 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

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

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, because 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:

$$\{(X_1, G_1), \ldots, (X_n, G_n)\}$$

We propose two procedures: one for 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 we have 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 command `mstat` or `mtest`. We propose the following algorithm:

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

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

   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; that is, multiply by $k \geq 100$ (for instance, 500 or 1,000, 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 because of 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 and 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. The test is based on equally 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, because 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. We provide a listing and a graph (figure 3) of the data:

```
. 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) density
```

```
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 confidence interval with respect to p=c/n
```

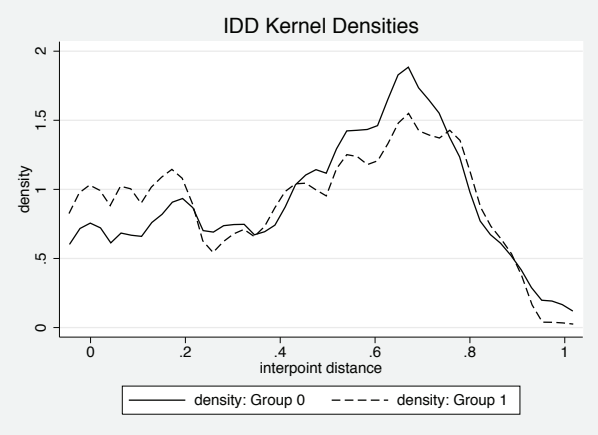

Figure 3. Kernel densities for groups 0 and 1 in the large-sample setting with discrete data.

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:

1) A dataset resulting from the composition of breast cancer data extracted from Massachusetts Department of Public Health (2009) with census tracts coordinates for the State of Massachusetts

2) Scott Merryman's U.S. county map coordinates for Pisati's package

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" (Massachusetts Cancer Registry 2009). We focus here on breast cancer (only among females), a cancer whose site (and gender) are 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

$$\texttt{rel}_s = \frac{\text{observed}_s \;-\; \text{expected}_s}{\sqrt{\text{expected}_s}},\;\; s = 1, \ldots, 348$$

a standardized difference between observed and expected numbers of cases.

The variable `rel` is approximately normally distributed with a zero mean, which 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

$$G_s = \mathbf{1}\left(\texttt{rel}_s > 0\right), \ s = 1, \ldots, 348$$

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.

We can also use the dataset together with Pisati's `spmap` package (Scott Merryman's [2005] polygons data) to map the distribution of G. We report here the do-file for executing the test on the data and its results. Instead of calling the `scatter` option, we use `spmap` to superimpose variable G on the map of Massachusetts together with the locations of the 31 Environmental Protection Agency Superfund sites in the state, as shown in figure 5.[5]

```
. use breast_cancer.dta, clear
. mtest, x(Long) y(Lat) g(G) iter(1000) density dlabel0("obs<exp")
> dlabel1("obs>exp")
> dtitle(IDD of standardized relative incidence for breast cancer in MA)
```

```
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 confidence 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.

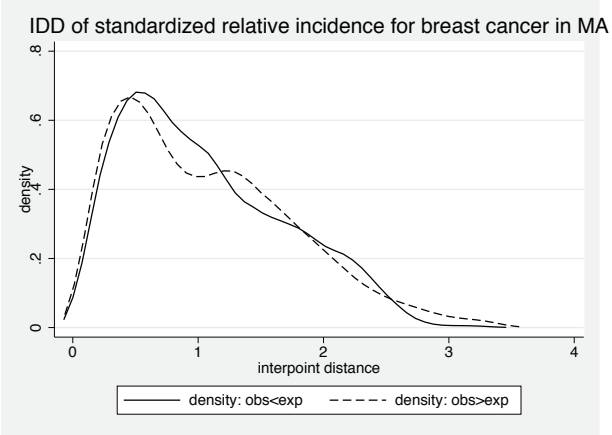

Figure 4. Kernel densities. Units in group 1 are locations with observed cases exceeding the expected.

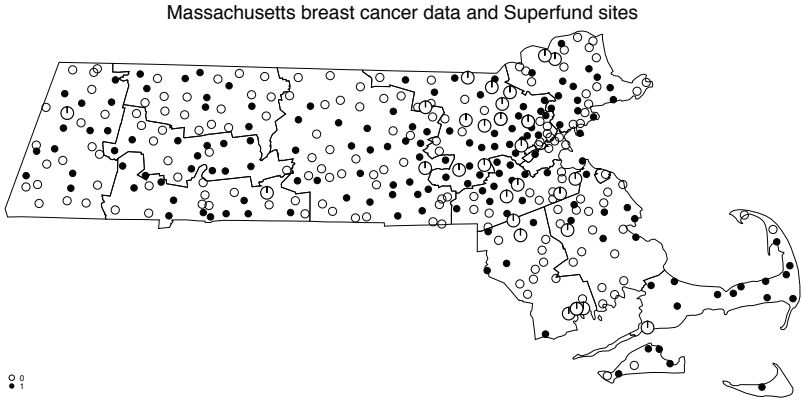

Figure 5. Map of Massachusetts. The black dots correspond to locations with observed cases exceeding the expected. Empty dots correspond to locations with observed cases not exceeding the expected. The circles around some of the locations indicate the presence of an Environmental Protection Agency Superfund 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, as shown in figure 4.

5 mstat and mtest: Syntax, options, and saved results

The syntaxes for the two commands are the following:

`mstat, x(`*varname*`) y(`*varname*`) g(`*varname*`)` [*mstat_options graphic_options*]

`mtest, x(`*varname*`) y(`*varname*`) g(`*varname*`)` [*mtest_options graphic_options*]

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

5.1 Options for mstat

`bins(#)` selects the number of bins. # must be a positive integer that is not larger than the number of distances in the dataset (that is, the number of observations). The default is `bins(20)`. For the theoretical implications, refer to White, Bonetti, and Pagano (2009).

`chi2` displays and returns the p-value for the asymptotic chi-squared test (upper tail).

`scatter` generates a scatterplot of the two groups.

`density` generates a kernel density of the interpoint distance distribution for the two groups.

5.2 Options for mtest

`bins(#)` selects the number of bins. # must be a positive integer that is not larger than the number of distances in the dataset (that is, the number of observations). The default is `bins(20)`. For the theoretical implications, refer to White, Bonetti, and Pagano (2009).

`iter(#)` sets the number of random permutations of the group variable (the `g(`*varname*`)` option) to be performed for the Monte Carlo test. The default is `iter(100)`.

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

`scatter` generates a scatterplot of the two groups.

`density` generates a kernel density of the interpoint distance distribution for the two groups.

5.3 Graphic options

The graphic options are designed to manipulate the graphic output of the commands. The options are active only if the corresponding option (`scatter` or `density`) is specified.

Options when scatter is specified

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

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

`smarker0(`*symbolstyle*`)` sets the symbol for the marker of group 0.

`smarker1(`*symbolstyle*`)` sets the symbol for the marker of group 1.

`ssize0(`*markersizestyle*`)` sets the size for the marker of group 0.

`ssize1(`*markersizestyle*`)` sets the size for the marker of group 1.

`slabel0(`*string*`)` inputs the label for group 0 in the legend. The default is `slabel0("Group 0")`.

`slabel1(`*string*`)` inputs the label for group 1 in the legend. The default is `slabel1("Group 1")`.

`stitle(`*string*`)` specifies the title for the scatter. The default is `stitle("Spatial Distribution of the two groups")`.

`sytitle(`*string*`)` specifies the title for the y axis. The default is the name of the variable in option `y(`*varname*`)`.

`sxtitle(`*string*`)` specifies the title for the x axis. The default is the name of the variable in option `x(`*varname*`)`.

Options when density is specified

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

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

`dpattern0(`*linepatternstyle*`)` sets the pattern style for the line of the density of group 0.

`dpattern1(`*linepatternstyle*`)` sets the pattern style for the line of the density of group 1.

`dwidth0(`*linewidthstyle*`)` sets the width for the line of the density of group 0.

`dwidth1(`*linewidthstyle*`)` sets the width for the line of the density of group 1.

`dlabel0(`*string*`)` inputs the label for group 0 in the legend. The default is `dlabel0("Group 0")`.

`dlabel1(`*string*`)` inputs the label for group 1 in the legend. The default is `dlabel1("Group 1")`.

`dtitle(`*string*`)` specifies the title for the kernel density. The default is `dtitle("IDD Kernel Densities")`.

5.4 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 $M \geq M(\text{obs})$ 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, particularly in epidemiological studies. The one-sample version of the test requires the user to have knowledge of the null distribution, that is, the IDD that is compared with the observed one, or at least the density in each bin. Because this often is not available, except when we work with census data (for example, Bonetti and Pagano [2005]), 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-sample M test where the groups correspond to the two datasets, the only difference being that in the two-sample case when computing the matrix $\mathbf{S}$ we take into consideration the variability in both groups. The loop for the construction of this matrix $\mathbf{S}$ 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 only with datasets in the two-dimensional Euclidean space. Because 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 IDD in k-dimensional Euclidean spaces and, more generally, with generic dissimilarity measures. Alternatively, one could also develop a shorter version of the command to have the user inputting directly the dissimilarity matrices $\mathbf{D}$ (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 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 is 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 grants from the NIH, P01 CA 134294 and R56 EB006195.

8 References

Bonetti, M., and M. Pagano. 2005. The interpoint distance distribution as a descriptor of point patterns, with an application to spatial disease clustering. *Statistics in Medicine* 24: 753–773.

Manjourides, J., and M. Pagano. Forthcoming. Improving the power of chronic disease surveillance by incorporating residential history. *Statistics in Medicine*.

Massachusetts Department of Public Health. 2009. Cancer Incidence in Massachusetts 2002–2006 Supplement. http://www.mass.gov/dph/mcr.

Merryman, S. 2005. USMAPS2: Stata module to provide US county map coordinates for tmap. Statistical Software Components, Department of Economics, Boston College. http://econpapers.repec.org/software/bocbocode/s448404.htm.

Pisati, M. 2004. Simple thematic mapping. *Stata Journal* 4: 361–378.

Waller, L. A., and C. A. Gotway. 2004. *Applied Spatial Statistics for Public Health Data*. Hoboken, NJ: Wiley.

White, L. F., M. Bonetti, and M. Pagano. 2009. The choice of the number of bins for the M statistic. *Computational Statistics & Data Analysis* 53: 3640–3649.

About the authors

Pietro Tebaldi completed his master of science studies in economics at Bocconi University and worked as a visiting scientist in the Biostatistics Department at the Harvard School of Public Health during the summer of 2010. He will start his doctoral studies in the Department of Economics at Stanford University in the fall. He is interested in economic theory and in the development of empirical methods for economic policy analysis.

Marco Bonetti is an associate professor of statistics in the Department of Decision Sciences at Bocconi University. Previously, he worked as a cancer clinical trial biostatistician and as an assistant professor of biostatistics at the Harvard School of Public Health and the Dana–Farber Cancer Institute. His research interests are in biostatistics and, in particular, in biosurveillance and in the analysis of clinical trial data.

Marcello Pagano is a professor in the Biostatistics Department at the Harvard School of Public Health. He has been there for the last 33 years. His research interests in biostatistics are sufficiently varied such that he will never be sated.