The *CrimeStat* Program: Characteristics, Use, and Audience

Ned Levine, PhD
Ned Levine & Associates and Houston-Galveston Area Council
Houston, TX

In the paper and presentation, I will discuss the *CrimeStat* program, its potential uses, and the audience for whom it is intended. In addition, I will also comment on the need for transparency among statistical software, but will distinguish between clear documentation, open data and conversion standards, and open source code.

The *CrimeStat* Program

CrimeStat is a stand-alone spatial statistics program for the analysis of crime incident locations that can interface with most desktop GIS programs. It was developed by myself and Long Doan under research grants from the National Institute of Justice. The program is Windows-based and interfaces with most desktop GIS programs. The purpose is to provide supplemental statistical tools to aid law enforcement agencies and criminal justice researchers in their crime mapping efforts. The National Institute of Justice is the sole distributor of *CrimeStat* and makes it available for free to analysts, researchers, educators, and students.[1] The program includes a manual/textbook that describes each of the statistics and gives examples of their use. The manual is also available for download.

The program is written in C++ and is multi-threading. It will take advantage of multiple processors in a computer which, for a large data set, will considerably cut down on calculation time. The program also includes Dynamic Data Exchange code to allow another program to call up *CrimeStat* and pass the dataset name and variable parameters to it. Such a use was developed by the Criminal Division of the U.S. Department of Justice in developing the Regional Crime Analysis GIS (RCAGIS). That application used the Internet to link weekly crime databases from jurisdictions in the Baltimore metropolitan area to a common interface and set of analytical tools. *CrimeStat* was one of the tools.

CrimeStat is being used by many police departments around the country as well as by criminal justice and other researchers. From what we can tell, it has been used in many courses and has been a tool in a number of Masters and PhD theses. Three versions have been released. The first (1.0) was released in November 1999 and an update version was released in August 2000. The new version is 2.0 and will be released during the spring.

Data Input and Output

---

[1] The program is available at:

http://www.ojp.usdoj.gov/cmrc (under 'Mapping tools') or
http://www.icpsr.umich.edu/NACJD/crimestat.html

The program inputs incident locations (e.g., robbery locations) in 'dbf', 'shp' or ASCII formats using either spherical or projected coordinates. It can also treat zones as pseudo-points (or points with intensities). The program calculates various spatial statistics and writes graphical objects to ArcView®, ArcGis®, MapInfo®, Atlas*GIS™, Surfer® for Windows, ArcView Spatial Analyst©, as well as programs that follow the ODBC standard.

Program Sections

CrimeStat is organized into five sections:

Data Setup

1. **Primary file** - this is a file of incident or point locations with X and Y coordinates. The coordinate system can be either spherical (lat/lon) or projected. Intensity (Z) values and weight values are allowed. Each incident can have an associated time value.

2. **Secondary file** - this is an associated file of incident or point locations with X and Y coordinates. The coordinate system has to be the same as the primary file. Intensity and weight values are allowed. The secondary file is used for comparison with the primary file in the risk-adjusted nearest neighbor clustering routine and the duel kernel interpolation.

3. **Reference file** - this is a grid file that overlays the study area. Normally, it is a regular grid though irregular ones can be imported. *CrimeStat* can generate the grid if given the X and Y coordinates for the lower-left and upper-right corners. Several routines utilize the reference file.

4. **Measurement parameters** - this identifies the type of distance measurement (direct or indirect) to be used and specifies parameters for the area of the study region and the length of the street network.

Spatial Description

5. **Spatial distribution** - statistics for describing the spatial distribution of incidents, such as the mean center, center of minimum distance, standard deviational ellipse, Moran's I, Geary's C, or the directional mean.

6. **Distance analysis** - statistics for describing properties of distances between incidents including nearest neighbor analysis, linear nearest neighbor analysis, and Ripley's K statistic.

7. **'Hot spot' analysis I** - routines for conducting 'hot spot' analysis including the mode, the fuzzy mode, hierarchical nearest neighbor clustering, and risk-adjusted nearest neighbor hierarchical clustering.

8. **'Hot spot' analysis II** - more routines for conducting hot spot analysis including the Spatial and Temporal Analysis of Crime (STAC), K-means clustering, and Anselin's local Moran.

Spatial Modeling

9. **Interpolation** - a single-variable kernel density estimation routine for producing a surface or contour estimate of the density of incidents (e.g., burglaries) and a dual-variable kernel density estimation routine for comparing the density of incidents to the density of an underlying baseline (e.g., burglaries relative to the number of households).

10. **Journey to crime analysis** - a criminal justice method for estimating the likely location of a serial offender given the distribution of incidents and a model for travel distance.

11. **Space-time analysis** - a set of tools for analyzing clustering in time and in space. These include the Knox and Mantel indices, which look for the relationship between time and space, and the Correlated Walk Analysis module, which analyzes and predicts the behavior of a serial offender.

Options

12. Parameters can be saved and re-loaded.

13. Tab colors can be changed.

14. Monte Carlo simulation data can be output.

CrimeStat is accompanied by three sample data sets and a manual that gives the background behind the statistics and examples.

Audience

Any program has to have an audience to whom it is addressed. Crudely, a distinction can be made between four audiences for whom a statistical package would be appropriate, recognizing that in reality there are always mixtures of the four:

1. Statisticians
2. Researchers
3. Students
4. Analysts

CrimeStat was developed primarily for researchers and analysts, and secondarily for students. It was aimed at providing statistical tools to allow criminal justice researchers and analysts to quickly identify patterns in the distribution of incident locations. This is important in crime analysis. Thus, an emphasis is placed on pattern identification while

statistical significance is treated as a secondary issue.

The program does have a number of formal statistical tests and includes six different routines where a Monte Carlo simulation can produce an approximate statistical test. Nevertheless, the emphasis was placed on graphical representation that can be displayed on a GIS. The integration of *CrimeStat* with a GIS package is an essential part of the program. The underlying philosophy behind the program is as a GIS-based tool to help researchers and analysts in their work. It also differs from SAS, SPSS, S-Plus, and other statistical programs in that it has a detection and identification philosophy that underlies it. No attempt has been made to produce a comprehensive collection of tools. Instead, the tools that have been included were chosen because they are useful to crime analysts and criminal justice researchers (plus, hopefully, others).

Among the uses of the program are hot spot identification (i.e., clusters of crimes that concentrate), visualizing temporal shifts in spatial patterns, the identification of crime 'risk', and the analysis of serial events (e.g., a serial offender). There are many others uses for which the program can be, and has been, used, but those are the primary ones. For a police department, hot spot classification helps in the allocation of police officers. The recognition of temporal shifts in crime allows a more flexible police and community response. The identification of high risk areas is useful for crime prevention purposes while the analysis of serial events is important in apprehending dangerous offenders.

Transparency

Finally, I will provide some thoughts on transparency in statistical software development based on five years of experience in developing this software package.

Clear Documentation

I believe that open and clear documentation is essential for the development of knowledge by others. It's important for a software developer to provide as clear and comprehensive information as possible on the algorithms and methods used in the program. Only if users clearly understand what a routine does will they be able to properly use it. Too often, 'homemade' software has very cryptic documentation. The result is that it is more likely to be ignored than used.

Common Data and Conversion Standards

Similarly, I believe that common standards are essential for the widest development of GIS and spatial information systems. Common data standards allow data to be converted and passed from one program to another. Common program 'hooks' allow third-party developers and researchers to write new routines that can be used by a number of software programs. Ideally, if every statistical software developer or manufacturer used a common set of communication links, then small developers could see their routines used by a variety of users on different program platforms. Common standards can allow users to tailor their analysis to what they need to accomplish, knowing that they can take advantage of a variety of tools, rather than have to rely on those made available by a single producer. Finally,

from an economic perspective, common standards allow many producers to emerge and encourage creativity and innovation.

Open Source Code

On the other hand, I have doubts about the value of open source code. On the one hand, making code open could improve it since it would no longer just be the province of a single producer. But, it could also lead to abuse and breaches of security that could be dangerous to a wide range of users. Computer code, particularly low-level languages, can be very cryptic and personal. Different programmers have their own style and it may take a new programmer a long time to figure out that style. Variations on the original code could turn out to be wrong. In theory, if there are lots of people looking at the code, one would expect a greater 'wisdom' to emerge out of the collective efforts than if a single producer only controlled the code. On the other hand, many program applications, particularly statistical ones, have a small audience; there are few programmers with enough knowledge or motivation to dig into someone's code. If a third-party programmer makes modifications that turn out to be wrong, people may use the routine thinking that it's right and it may take a while to discover that it is not.

Currently, developers produce and own the existing code. The good ones will periodically fix the problems that emerge and update the program. This process can lead to trust by users. With open source code, that trust may not emerge. There are also some security concerns about open code. It is too easy to hide viruses, 'trojan horses', and other types of destructive agents within a cryptic code. Again, unsuspecting users may stumble upon these destructive processes and have damage done to their data or storage media.

In short, while I'm not rejecting the idea of open source code, I believe there are some serious issues that need to be addressed before we fully commit to such a course.