Document Title: Making UCR Data Useful and Accessible

Author(s): Michael D. Maltz ; Joseph Targonski

Document No.: 205171

Date Received: April 2004

Award Number: 2001-IJ-CX-0006

Making UCR Crime Data Useful and Accessible

Michael D. Maltz and Joseph Targonski

Department of Criminal Justice
University of Illinois at Chicago

February 2, 2004

ABSTRACT

Crime data, collected by the FBI through its Uniform Crime Reporting (UCR) program, have been used to allocate Federal moneys and to study the effect of different policies on crime. Unfortunately, the data have many gaps that affect the accuracy of the analyses that have used the UCR. The nature and consequences of these gaps are not very well known to researchers, let alone legislators and the general public. There were three goals of this project: to clean the data set and account for the gaps, as well as to annotate the data to indicate the type of problem each gap represented; to make the data more accessible by combining the data from 1977-2000 into a single data set; and to develop and test methods of imputing the missing data.

The number and variety of problems we encountered in cleaning and combining the data prevented us from even starting to address the third goal, developing imputation methods. We used exploratory data analysis techniques (primarily visual) to pinpoint the data gaps and anomalies. We then categorized the gaps according to the type of "missingness" they represented – e.g., agency did not exist in that year, agency's crime data were reported through another agency, agency reported crime data for that month in a subsequent month, typographical error in entering the data, etc.

The product of our study is a set of 51 Excel files, 50 state files containing (annotated) crime and ancillary data and one plotting file that makes the data accessible. One can plot the data from 1977-2000 for every police agency, for the Crime Index, for violent and property crime, or for any one of the seven individual crimes that comprise the Crime Index. County-level crime data can also be plotted, although there are some inaccuracies present. The greater accessibility of these files should make subsequent studies of crime easier to accomplish, and the annotation of the types of missing data found in the files should improve the accuracy of such studies. It will also aid in the development of improved imputation methods.

I. Introduction

This grant had three overall goals: to clean the crime data collected annually by the Federal Bureau of Investigation (FBI) as part of its Uniform Crime Reporting (UCR) Program, to make the data more accessible by combining the data from 1977-2000 into a single data set, and to develop and test new methods of imputation to fill in gaps in the UCR data.

It has always been difficult to study crime patterns over time using the UCR files archived at the National Archive of Criminal Justice Data (NACJD). To study an individual agency's crime trajectory, it has been necessary to download each year's data set, convert it into an SPSS or SAS file, and merge it with the other years' data sets. We hoped that we would be able to combine the crime data from all years into a single data set, clean and annotate the data, and test and develop imputation algorithms to fill in the gaps in the data.

The impetus for this effort was twofold. First, I was a Visiting Fellow at the Bureau of Justice Statistics (BJS) in 1995, when the Local Law Enforcement Block Grant Act was passed. This law provided funds to police departments based on the amount of violent crime their jurisdictions had experienced, based on the departments' UCR violent crime statistics. In investigating the characteristics of the UCR crime data, a number of problems were uncovered; they are described in the report, *Bridging Gaps in Police Crime Data* (Maltz, 1999).

Second, over the years NACJD has been aggregating UCR crime data to the county level and publishing them on-line. There were (and are) many problems with this aggregated form of the UCR that were not apparent; however, since county-level data (if valid) can be very useful and are being used to analyze policy issues (Lott, 1998, 2000; Maltz & Targonski, 2002), we wanted to improve the quality of the county-level aggregation of UCR data as well.

In retrospect, it turns out that we were overly optimistic about our ability to expeditiously accomplish the first two goals, cleaning and combining the data, and go on to the second, developing imputation methods. In particular, despite our prior knowledge of the UCR's deficiencies, we were surprised by their extent. Much of our work relied on hands-on methods of spotting problematic data points and coding them according to the nature of the problem, so the more problems there were, the longer it took us to clean the data. It is for this reason that we were unable to accomplish the second goal.

We have, however, completed the first two goals and have prepared a set of annotated state data files. In addition, we provide a plotting utility so that the monthly crime data for any agency or county can be plotted. In other words, we have provided the means to review *the entire crime history from 1977-2000 for virtually every police agency in the United States, for each of the 7 Index crimes*[1] *and their total, as well as a limited look at county crime data for the same period*. Although these data have been

[1] They are: murder, forcible rape, robbery, aggravated assault, burglary, larceny, and auto theft. Arson was added to the Crime Index in 1979; it is not as likely as the other Index crimes to be reported to the police, because arsons are often categorized as "fires of suspicious origin." The FBI includes it in the "Modified Crime Index," but not the more popular Crime Index.

available for some time, this project has put them in a format that makes them readily accessible to anyone.

The amount of information we dealt with is extensive, especially considering the fact that we used visually inspection to detect anomalies in the data. We reviewed 24 years of crime data (1977-2000 inclusive); with 12 monthly reports for each year for each of the 7 Index crimes; for every agency that provides crime reports to the FBI, this latter amounting to over 18,000 agencies (or, in the FBI's parlance, **ORI**s, for ***OR***iginating Agency ***I***dentifiers). In other words, we have been engaged in reviewing, inspecting and cleaning some 36 million data points. While we have employed automated means of doing so, our first approach was visual, to see the extent to which suspicious and anomalous patterns could be found in the data.

This report describes the process we went through to prepare the UCR crime data. The next section describes the sources of the data we used to prepare our state UCR files. Section III discusses the software we used for preparing the crime data. The missing value codes we used are detailed in Section IV. We detail the specific procedures we used to clean, analyze, and plot the data in Section V. Finally, Section VI provides a list of additional tasks to be undertaken that will provide additional enhancements to the monthly data series of Crime in the United States.

II. Data Sources

The data we used were obtained primarily from the National Archive of Criminal Justice Data (NACJD).[2] NACJD archives the raw crime data from the UCR program; 1966 is the first year archived, but the full archive (including "Offenses Known," "Clearances by Arrest," "Property Stolen and Recovered," "Supplementary Homicide Reports," "Police Employee (LEOKA) Data," and "Hate Crime Data" for 1992 and beyond) started in 1975 – prior to then only "Offenses Known to the Police" and various offense supplements were included (NACJD, 2004). We worked with the "Offenses Known" data [ICPSR file 9028] for the 24 years from 1977 to 2000.

The NACJD website that contains the raw crime data also contains syntax files to permit users to convert the data into SPSS or SAS data sets. We converted the data files into SPSS format. As anyone who has used these files knows, the variable names are opaque (e.g., "v101" instead of "January 1995 Murders"). Our first task was to write syntax files to make the names more understandable, and so that we could combine the annual files to make one large master file. Since SPSS variable names are limited to eight characters, we renamed them using the following structure: "MU.95.01" represented the number of murders reported for January 1995, and the other crimes were designated "RA," "RO," "AS," "BU," "LA," and "MV," with "CI" for their sum, the Crime Index. Each year was then merged to create on master file in SPSS, with all monthly agency data from 1977-2000.

A word of caution: we used "00" to signify the year 2000, despite the (Y2K) confusion this caused when the new century began. We do not feel that this will cause any problems in this case, since the digits are used as (text) indicators and will not be put in numerical order.

It turned out that the 1994 data set that we had obtained from NACJD was corrupted, at least for some of the variables of interest for our study. Apparently no earlier user of the data set had noticed this problem, which is probably due to the fact that other users had more limited goals in their studies. After working with NACJD personnel to determine the nature of the problem, we concluded that the raw data they had archived had been in error. We subsequently received a new version of the 1994 data from the FBI, as did NACJD.

Extent of Data Used. The UCR raw data files include many different types of data. They include: ORI name; address; county (in some cases, where agencies straddle county borders, multiple counties); population residing in each county; SMSA indicator; population group; crimes by type (with some disaggregated by subtype); attempted crimes by type; date the information was updated; agency reporting the ORI's data (if the agency didn't report its data by itself); etc. We did not use all of this information; for example, since we are focusing on completed crime, we ignored attempts, and we did not work with the disaggregated crime data.

[2] We had originally obtained the crime data from the National Consotrium on Violence Research (NCOVR), which they had archived in an Oracle database. We subsequently found it more practical to use the NACJD data.

Moreover, we used some of the data in ways that had not been anticipated by the FBI. For example, if the ORI was missing from a data set, we interpreted it to mean that the agency did not provide data for that year. For the most part, ORIs were missing either for the first few years (prior to the agency's existence) or for the last few years (after the agency no longer existed, possibly having been absorbed into another agency). If an agency was in the data file for that year, so was its name. Therefore, the ORI name was used to provide an indication of (in demographic terms) agency "birth" and "death."

As part of our data cleaning process, we manually examined any ORI that changed names over the 24-year period. This allowed us to distinguish any data entry errors that may have occurred with the ORI code.

"Date updated" is another variable that was used as an indicator for something else for which it was not intended. The FBI uses it to indicate when the report for that month was received. If no "date updated" exists for a particular month, and there are no entries for crime for that month, we interpreted that as meaning that the zeros in the crime columns represented missing data, not zero crime. [Otherwise there is no way of distinguishing a true zero from missing data.]

We also noted problems in the opposite direction. That is, we found cases where there was no "date updated," but the agency actually had reported crime data.

The "date included in" variable is a flag the FBI uses to indicate when an agency's crime data for a month is included in a different month's data. For example, an agency that reports data on a semiannual basis should have a flag for months January through May, signifying that the data will be aggregated into the June data. However, we found this variable to grossly underestimate such data aggregation and we do not suggest relying on it. Our visual inspection of the data was designed to catch such data issues and anomalies.

We included additional data that others might find useful. These include the SMSA designator and the FBI's population group category.

In retrospect, we should have included more population data than we did. That is, the UCR data sets allow for the possibility that a jurisdiction may spread across up to three counties. We retained only the total population for the jurisdiction, when we should have included all three population figures and county designators. This would have permitted us to provide a more accurate estimate of county-level crime than we now have.

III. Software Used

Although we had converted the data to SPSS, that database software product does not have the versatility that we needed for our analyses. Its plotting routines are limited, as is the user's ability to program macros. We used SPSS primarily to organize the data preparatory to exporting it to Excel. The SPSS syntax file we used to export the data is given in Appendix A.

We also wanted to use color to represent different characteristics of a datum. For example, a datum (a particular ORI's crime count for a given month and given crime) might be missing because it wasn't reported (red); because the ORI didn't exist at the time (blue); because its crime count was combined with another agency's (cyan); or because it was aggregated to be reported quarterly, semiannually, or annually (green). See Table 1 for a listing of the colors and codes.

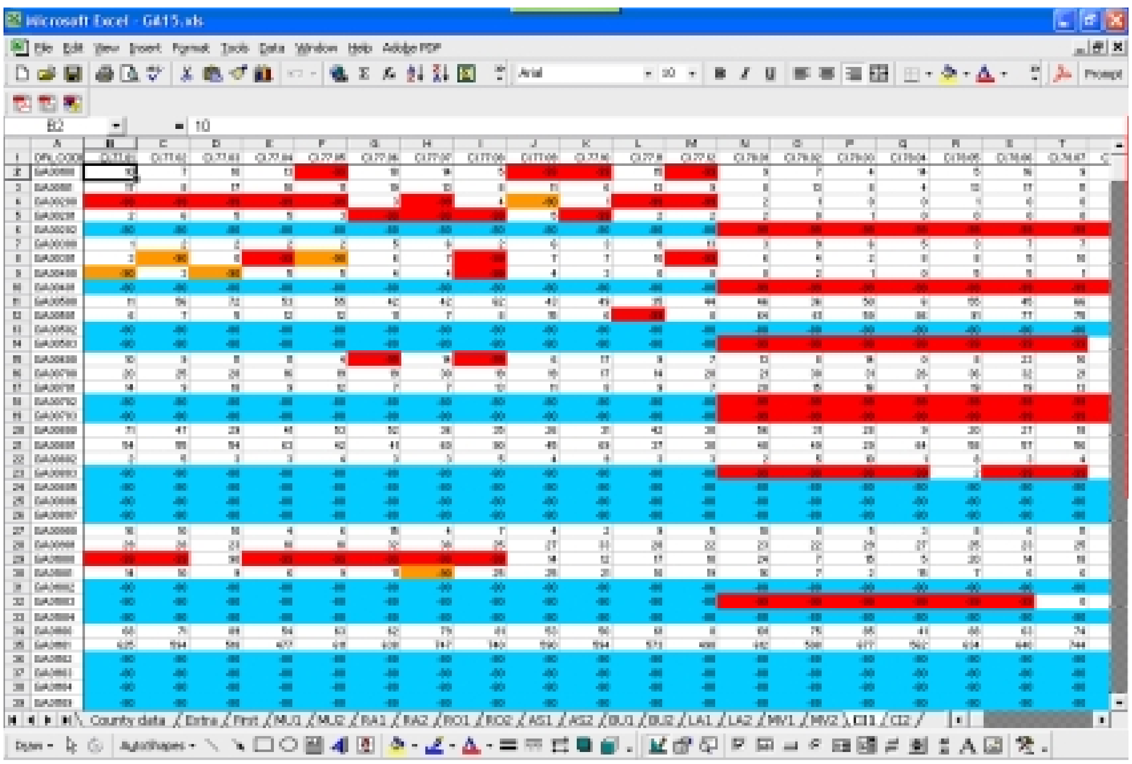
Coloring the cells permitted us to get an overall indication of the nature of the crime data for that particular jurisdiction. That is, when the entire worksheet is depicted,

Code	Value	Color
agency did not exist during this period	-80	blue
ORI is covered by another agency	-85	cyan
we assign missing and record its value	-90	orange
murder missing	-91	red
rape missing	-92	red
robbery missing	-93	red
assault missing	-94	red
burglary missing	-95	red
larceny missing	-96	red
motor vehicle theft missing	-97	red
on CI page, more than 1 crime missing	-98	red
no data for this month (true missing)	-99	red
aggregated to February	-102	green
aggregated to March	-103	green
aggregated to April	-104	green
aggregated to May	-105	green
aggregated to June	-106	green
aggregated to July	-107	green
aggregated to August	-108	green
aggregated to September	-109	green
aggregated to October	-110	green
aggregated to November	-111	green
aggregated to December	-112	green

Table 1. Color Codes for Missing Values

one can see the extent to which missing values, data aggregation, agency "births" and "deaths," and agencies reporting through other agencies occur. For these reasons, we decided to use Microsoft Excel as the primary software in our analysis. An additional advantage is that Excel includes Visual Basic for Applications, giving us the flexibility to search for different patterns in the data and make corrections according to the need. Figure 1 is a screen shot of a state file, showing the patterns and types of "missingness."

Figure 1. A Screen Shot of a Portion of a State File



For example, an agency that did not exist in 1977 would show up in our combined (1977-2000) file without anything in the "NAME.77" column for that year. It occurs when a city has not yet grown to the point where it had its own police department; or while a university police department is considered, for reporting purposes, an adjunct of the police department in which it is embedded; or when a suburb is absorbed into a larger city. Once we found the years when an ORI did not exist, we could use a Visual Basic macro to find all the cells representing those years and color them all blue (and at the same time changing their values from 0, the default value for the FBI data, to –80, the value we chose to represent a nonexistent ORI).

Unfortunately, there are also some limitations to Excel, for which we devised work-arounds. The primary limitation was in the number of columns, a maximum of 256. It was for this reason that we had to use two worksheets to tabulate all of the data for each individual crime (24 years x 12 months = 288 columns). This same limitation affected our ability to make simple graphs, since the entire sequence of data points stretched across two worksheets, complicating the graphing process somewhat.

Also, the size of the state spreadsheet files is large compared to those for database files. This is due directly to the versatility of the spreadsheet: each cell in a spreadsheet can have a number of attributes (color, font, comments, etc.), and can contain text, formulas, or numbers; in contrast, every cell in a column of a database must consist of the same type of data, either numbers or text.

We used a series of Visual Basic macros to determine the nature and extent of different anomalies. They are listed in Appendix B.

IV. Annotating Missing and Questionable Crime Counts

We originally used negative decimal values (i.e., between –0.01 and –0.99) to indicate missing and questionable crime counts. We decided upon this because negative numbers of crimes do occur, as crimes reported in one month are unfounded in a later month. And, of course, there are no non-integer numbers of crimes, so it would be easy to distinguish between negative decimals, representing missing values, and negative crimes.

However, colleagues informed us that certain software packages might have problems with this type of assignment, so we revised our coding procedure. In their place we used large negative numbers, between –80 and –112, to represent the different types of missing data we encountered. We used such large numbers (in absolute terms) because we encountered many (apparently) *true* negative numbers, as when crimes occurring in one month are unfounded in a subsequent month; in this way we assured separation between the true negatives and the missing values.

Some of the reported negative numbers were quite large: the largest negative value was –163. Obviously, this is a mistyped number; in fact, we assumed that the largest negative value that might be valid was –3, using the following logic: if an agency had 4 or more crimes that it coded as unfounded, it probably experienced on average at least 10 crimes per month. And to have a net of –4, it would have to have experienced at least 2 reported crimes and 6 unfoundeds, which seemed to us to be stretching the limits of credibility. The number of negative crime counts we found is given in Figure 2.

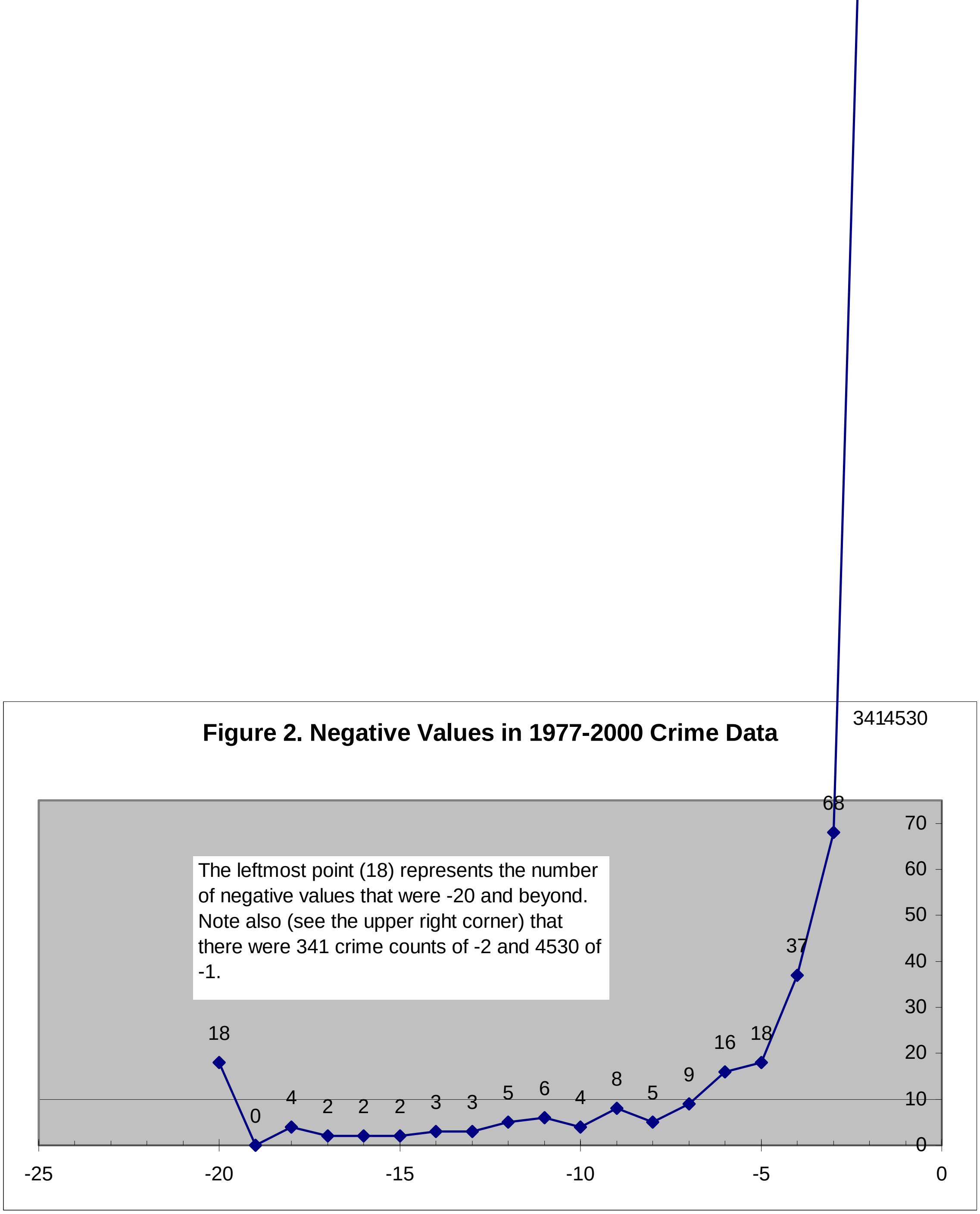


Figure 2. Negative Values in 1977-2000 Crime Data

The data contained 5081 negative values for monthly crime counts, out of the over 40 million values in the data set, a very small number (.01 percent). Of these, only 142 were larger (in absolute terms) than –3, and all of these were recoded as missing values.

Excel includes useful functions, like SUMIF, which permits one to find the sum of a number of cells that satisfy a given condition. That is, SUMIF(A1:A100, “>-4”) will add all of the values in the cells A1 through A100 that are greater than -4. This permitted us to exclude all of the missing data from our calculations of total crime.

We also used this type of procedure when plotting the crime data. Excel applies default values to the plots that are generated, based on the maximum and minimum values in the plotted data. However, we set the vertical axes of all of the plots to start at zero rather than at the smallest values, since these might be –90 or so if the plotted data included missing values.

V. Procedures Used to Clean the UCR Data

Before describing the manner in which we cleaned the data, it should be understood that there are very clear limits on our ability to ferret out bad data. Obviously, we cannot tell if an ORI chooses to “bury” a certain number of crimes by not recording them. And problems like this continue to exist: in 2003 alone there have been reports of burying crime in New York City, Philadelphia, and New Orleans. There is no conceivable way that we can deal with this data problem.

Some of these data anomalies, however, may be found by graphing the data over time. In some cases there were major and abrupt changes in crime reporting, as when the monthly crime count doubled or halved in a one- or two-month period and remained around the new level for some time. It would be of interest to determine the extent to which such major changes in crime counts were coincident with the change of a police policy or administrator, or of a municipal administration.[3]

As mentioned earlier, we used graphical techniques to do the bulk of the data cleaning. The overall procedure was generally as follows:

1. Obtaining and Aggregating Annual Crime Data Files. For each of the raw data files (i.e., each year) we applied its SPSS syntax to import it into SPSS. Since these files are quite large (there are over 1400 variables for each of the over 18,000 ORIs), we trimmed them to make them less cumbersome, by selecting only the variables we needed.

We selected the following categories of variables: monthly crime counts, for each Index crime; agency names, since they occasionally change (and are also useful in determining whether the ORI existed in that year); population in the primary county (some cities, and therefore agencies, cross the borders of counties);[4] whether the ORI’s data was reported by another agency for that year, and the ORI designation for that agency; the SMSA designation for that agency, in case other researchers want to look at the statistics of SMSAs; the primary county designation (the one for which we recorded the population),[5] using the FBI’s county codes;[6] and population group (see the FBI’s annual *Crime in the United States* or Maltz, 1999, p. 21).

In addition, we made use of the variable “date updated:” if it was blank for a given month, we interpreted it to mean that no data existed for that month.[7] This

---

[3] This issue would be a greater problem in dealing with arrest data than crime data, since a new police policy can have a greater effect on numbers of arrests than on numbers of crimes.

[4] In retrospect, we should have included the population figures for the other counties as well. We will return to this issue in Section VI.

[5] In some cases the primary county designation changed from year to year. In a few cases this had to do with the creation of a new county (La Paz in Arizona), but in most other cases we assumed that it reflected the changing distribution of population over time within the jurisdiction, with greater population accruing to a formerly non-primary county.

[6] The FBI’s county codes can be correlated with the Census Bureau’s FIPS codes using the Bureau of Justice Statistics (BJS) crosswalk file; see http://www.ojp.usdoj.gov/bjs/pub/pdf/lucrdod.pdf. To obtain the file, go to the website of the National Archive of Criminal Justice: www.icpsr.umich.edu/NACJD and click on “Download Data,” then enter study number 2876 in the box on the left and click on “Enter.”

[7] We verified this interpretation with FBI officials knowledgeable in UCR data elements.

permitted us to distinguish between a true zero (i.e., no crime occurred that month) and a zero representing missing data. We used this in two ways:

- We created a variable "RPTD.XX" (where "XX" is the 2-digit designation for the year). This variable shows the number of months in XX in which there was a non-blank "date updated," which is the number of months crime data were reported.

- Subsequently we decided that we needed to determine for *which particular months* crime was reported, so we added the variable "RPTMO.XX," containing a 12-character string of 0s and 1s, 0 designating "no report" for that month and 1 designating "crime reported" for that month.

2. Converting SPSS Files to Excel Files. We then combined all of the (trimmed) annual UCR files into one large SPSS file. We disaggregated the data by state and created fifty separate state Excel files.[8] The crime data were put on 16 worksheets, 2 for each crime type and 2 for the sum of the crimes. As mentioned earlier, two worksheets were needed for each crime type (and their sum) because we were dealing with 24 years of monthly data (or 288 monthly data points) for each ORI, and each worksheet has only 256 columns.

An additional worksheet contains the following variables: name, population, date updated, county number, SMSA indicator, population group, whether the ORI was "covered by" (i.e., reported through) another ORI, the ORI's (non)reporting history, and other such information.

A second worksheet was added to contain information related to plotting, finding and recording errors, recording changes in data entries, and other information used in our analyses. These worksheets (one for each state) are not included in the final version of the data sets, but are available from the principal investigator.

3. Finding Anomalies in the Crime Data. A macro was written to plot monthly (Index) crime data for each ORI for the 24 years. In this way we were able to inspect all 288 data points at once. Another macro was written to permit us to select, for the currently plotted ORI, a particular month of data to be the subject of a second plot. The second plot displayed the crime count for each of the seven Index crimes. If an anomaly was found (e.g., a month with a "spike" in the data[9]), we could use this macro to determine which crime accounted for the anomalous behavior. We then recorded the month and type of crime and the nature of the anomaly.

Each agency's crime count history was plotted and inspected visually. Going through some 18000 agencies took a number of weeks to complete, but we maintain that this procedure is more accurate than using an algorithm. We could judge, for example,

[8] At the moment we do not include the statistics for Washington, DC, Puerto Rico, Guam, or any other non-state jurisdiction.

[9] This occurred due to typographical error and due to interpreting "missing value" codes as data. Users of SPSS and SAS are familiar with coding missing values as 999; in some cases we found those codes in the data, but they were often treated as a crime count of 999 rather than a missing value. When an ORI that averaged 3 rapes per month suddenly showed up with 999 rapes, we considered this a missing value.

the likelihood that a zero was a true zero or merely a missing datum,[10] based on the volatility, seasonality, and trend of the time series as well as on the level of crime. An algorithm that was based on the level of crime, its average slope, its seasonality, and its variance (and how the variance changed over time) would be difficult to develop – and in the final analysis could only be validated by crosschecking it visually. We used a rule of thumb for determining if a zero crime index point was indeed a true zero: if an agency averaged at least 20 index crimes per month, and then the crime count in one month dropped to zero, that month would normally be coded as a true missing value.

In addition, we encountered a number of cases with extremely high numbers of individual crimes in one month. These cannot be eliminated algorithmically, since there have been cases of spikes in the number of single crimes in a single month (e.g., the bombing of the Murrah Building in Oklahoma City on April 19, 1995), so we investigated each one. In most cases the spike was due to entering a number that many database programs use as a missing value code (e.g., 999 or 9999); in those cases we replaced this value with the appropriate missing value color and code (i.e., red and -92 for rape; see Table 1).

4. Accounting for Anomalies in the "Covered-By" Data. An agency is "covered by" another agency if it doesn't report its data to the FBI (or state[11]) but reports through the other ("covering") agency. For example, a small town may not want to go to the expense of developing its own UCR reporting mechanism and will send its crime reports through the county sheriff's office.

If the agency is covered by another agency, then its crime data are not missing but are reported through the covering agency. The covered agency's crime trajectory would be zero during the time it was covered by the covering agency, and the covering agency's crime trajectory would be increased by the amount contributed by the covered agency.[12] Since both agencies are almost always in the same county, there should be no abrupt changes in the county's crime trajectory due to covered-by operations.

Covered-by data were missing from the raw data files for 1980 and 1995. We accounted for these gaps by inspecting each agency's covered-by status in the previous and subsequent years. That is, if an agency was covered by the same covering agency in 1979 and 1981, and reported no crime for 1980, we assumed that it was covered by the same agency for 1980; and similarly for the 1995 gaps.

Moreover, in some years and for some states the ORI identifier for the covering agency was longer than 7 characters.[13] To deal with this situation we truncated the ORI

---

[10] It was not always the case that a missing datum was signaled by the absence of the "date updated" field in the raw data. This meant that we had to plot the data to determine the extent of missing data.

[11] In most states, agencies do not report directly to the FBI but report through state criminal justice agencies, which then compile and transmit the data for the entire state to the FBI. See Maltz, 1999, p. 4.

[12] In most cases the crime contribution of the covered agency to the covering agency is relatively small, so one would not always expect to see an abrupt change in the covering agency's trajectory at the time the covered-by status changes.

[13] The syntax for an ORI identifier is SSCCCLLL, where SS is the two-character state designator (in upper case), CCC represents the (numerical) FBI designation for the county, and LL represents the two-digit designation for the local jurisdiction. So NB09101 represents Red Cloud in Webster County (FBI County No. 91), Nebraska. Also, note that the FBI uses NB for Nebraska instead of NE. NE is used by the US Postal Service and the Census Bureau, probably to eliminate confusion with New Brunswick, Canada.

identifier to 7 characters and checked to make sure that the resulting ORI existed. And in other years the ORI identifier for the covering agency was only 5 characters long. We used similar logic to infer the correct ORI automatically, but in ambiguous cases it was done manually.

Some of the problems we found in dealing with "covered-bys" are shown in the following figure and described below.

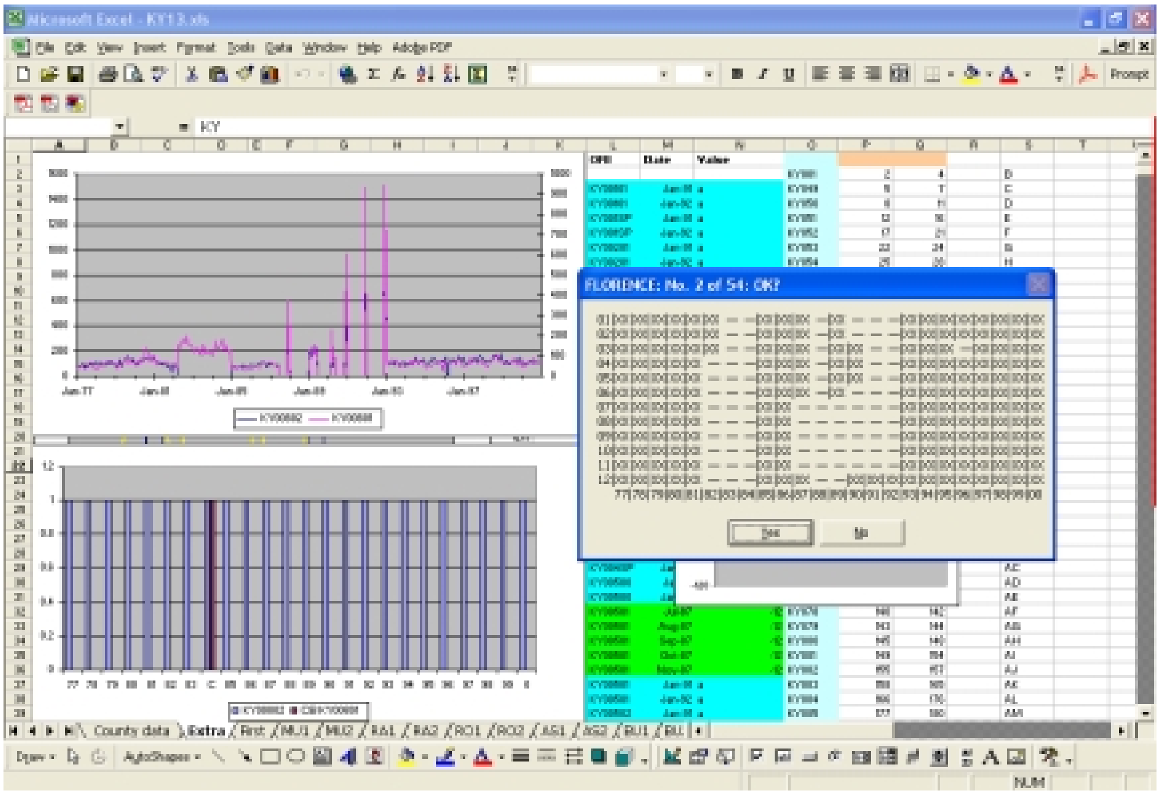


Figure 3. Screen Shot Showing Covered-By Anomalies

Figure 3 shows the crime for ORI KY00802, Florence, Kentucky. Start with the box on the right. The XXs show the months that Florence reported its own crime. In 1982 it reported crime for the first 3 months, and did not report crime for 1983 or 1984. In 1987 it provided crime reports for the first 6 months and for December. In 1988 there were no crime reports. In 1990 it provided reports for March-May and for December. And in 1991 and 1992 it provided crime data only for December.

Some of this is explained when we look at the number of monthly Index crimes reported (the blue line in upper left graph; its scale is the left axis). The reports provided by Florence in December 1991 and 1992 are obviously annual reports. Our best guess for 1990 is that the December report covers June-December crimes, and that January and February are missing.

The magenta line shows the monthly Index crime for its covering agency ORI KY00801, Boone County Police Department (right axis scale). It is obvious that the Boone County Police Department reported Florence's crime from March 1982 through December 1984.

The lower left chart shows Florence's "covered-by" status as reported in the UCR. It shows that Florence's crime data was reported as covered by the Boone County Police Department only in 1984, according to the UCR file. Note that the Boone County Police Department actually reported Florence's crime data for the last 9 months of 1982 and all of 1983, as well as for 1984.

From this example we can make a number of observations. First, if an agency reports being covered by another agency, it may not be for the entire year. Second, there are omissions in the UCR's "covered-by" reports. Third, it is sometimes difficult to determine whether a missing value is truly missing or whether it reflects being reported by another agency. This again demonstrates the advantage of checking crime covering visually.

5. Reporting Aggregate Data. Agencies do not always provide monthly reports. Some report quarterly, some semi-annually, some annually, and some may skip a month and report two months together. We developed macros that searched for these patterns and/or found them by inspection. Since these were not missing crime counts, they were treated differently and given different "missing value" codes. Even here we had to deal with some strange patterns. For example, from 1994-1996 Birmingham Alabama reported monthly data; however, the monthly counts were exactly the same,[14] leading us to conclude that the agency was actually reporting annual data that had been divided by 12. Chicago, Illinois, did the same (with quarterly data) in 1985.

6. Aggregating to the County Level. Since one of the factors motivating this research was the use of defective county-level crime data, we developed a macro to aggregate agency-level crime data to the county level. To this end, we decided not to use population data at all, although we include the Census Bureau's county population data on the state files. The reason for this is that we have noted some cases where the population is double-counted for some covered-by agencies (see Maltz & Targonski 2002).

There are some issues that we have not yet dealt with concerning county-level aggregation:

- Virginia has a large number of independent cities. We have not yet ascertained how best to deal with them.

- Similarly, we have not yet determined how to assign ORIs in Alaska to its boroughs.

- Many agencies are not assigned to single counties. They may have statewide jurisdiction, or they may have a missing county designator.

- Some agencies straddle county borders, and we need to deal with them as well. Such agencies' crime counts should be allocated to each county according to the percent of the population in each county.

---

[14] Or almost exactly the same; if there were 121 crimes in a year, 11 months reported 10 crimes and the 12th reported 11 crimes.

- Arizona added a county in 1982.
- Some agencies had no county codes.

VI. Conclusions and Recommendations

This project resulted in the creation of a (relatively) clean data set of crime in the United States from 1977-2000. The proof of the pudding in our case is the extent to which others will use it. To make the data set more accessible to its users, to facilitate its use in novel ways, we have created a crime plotting utility. It permits the user to plot an agency's (or county's) crime reports from 1977-2000, whether the Crime Index, violent crimes, property crimes, or a single crime. We anticipate that it will serve to provoke users to investigate anomalies in the crime data of individual agencies or counties. We also expect (actually, hope) that agencies will use it to examine their own data and inform us of any errors that may have crept into our data files.

This effort is just the first step in cleaning the UCR crime data, but we feel that additional steps can be taken to make the data more useful to potential users. They include:

- Accounting for the county-level problems described at the end of the last section.
- Adding to the utility the ability to copy the plotted data easily onto another spreadsheet.
- Porting the data to an SPSS or SAS file.
- Developing a sensible way to allocate statewide crime data to counties. In particular, we feel that, instead of allocating statewide crime data according to a county's population, it should be allocated according to a county's *unpoliced* population. For example, half of New York State Police crime data should *not* be allocated to New York City (because it contains about half the state population), but rather to the counties where they do the policing.
- Eliminating the spikes in crime data due to annual aggregation of data, so the plots can be more readily inspected.
- Depicting on the plot when an agency's data is missing and when it is being reported through (covered by) another agency.
- When plotting county data, the plot should provide an indication of the percent of the population that is represented in the data. It would also be useful to show on a map how this varies county by county, year by year. The state-by-state, county-by-county distribution of missing data is also worth depicting.
- Including the statistics for Washington, DC, Puerto Rico, Guam, and other non-state jurisdictions.

- Extending the series to the current year –and developing a set of procedures to incorporate each new year into the state files as the data become available.

REFERENCES

Lott, J. R., Jr. (1998). More Guns, Less Crime, University of Chicago Press, Chicago.

Lott, J. R., Jr. (2000). More Guns, Less Crime, second edition. University of Chicago Press, Chicago.

NACJD (2004). National Archive Of Criminal Justice Data. http://www.icpsr.umich.edu/NACJD/ucr.html#desc_al, accessed January 5, 2004.

Maltz, M. D. (1999). Bridging Gaps in Police Crime Data. Report No. NCJ-1176365, Bureau of Justice Statistics, Office of Justice Programs, U.S. Department of Justice, Washington, DC, September, 1999. http://www.ojp.usdoj.gov/bjs/pub/pdf/bgpcd.pdf.

Maltz, M. D., and J. Targonski (2002). "A note on the use of county-level crime data." Journal of Quantitative Criminology, 18, 3, 297-318.

Appendix A

SPSS Syntax for Creating Exel Files

Excel17.sps

This macro takes the final full SPSS file and creates 17 Excel files, one for the front (header) information, the other 16 for the crime data. A subsequent Excel macro (not included) combines the 17 Excel files into a single Excel file with 17 worksheets.

Note that the letters “XX” are replaced with the two-character state indicator, AL – WY, in 18 locations (the first to select the data for that state, the next 17 to create the Excel files).

```
GET
 FILE='C:\WINDOWS\Desktop\merged, 77-2000_2.sav'.
FILTER OFF.
USE ALL.
SELECT IF(substr(ori_code,1,2)="XX").
EXECUTE .
SAVE TRANSLATE OUTFILE='C:\WINDOWS\Desktop\XX_First.xls'
 /TYPE=XLS /MAP /REPLACE /FIELDNAMES / KEEP
state
ori_code
division
ag_state
name.77
name.78
name.79
name.80
name.81
name.82
name.83
name.84
name.85
name.86
name.87
name.88
name.89
name.90
name.91
name.92
name.93
name.94
name.95
name.96
name.97
name.98
name.99
name.00
pop4.77
pop4.78
pop4.79
pop4.80
pop4.81
pop4.82
pop4.83
pop4.84
pop4.85
```

```
pop4.86
pop4.87
pop4.88
pop4.89
pop4.90
pop4.91
pop4.92
pop4.93
pop4.94
pop4.95
pop4.96
pop4.97
pop4.98
pop4.99
pop4.00
cover.77
cover.78
cover.79
cover.81
cover.82
cover.83
cover.84
cover.85
cover.86
cover.87
cover.88
cover.89
cover.90
cover.91
cover.92
cover.93
cover.94
cover.96
cover.97
cover.98
cover.99
cover.00
rptd.77
rptd.78
rptd.79
rptd.80
rptd.81
rptd.82
rptd.83
rptd.84
rptd.85
rptd.86
rptd.87
rptd.88
rptd.89
rptd.90
rptd.91
rptd.92
rptd.93
rptd.94
rptd.95
rptd.96
```

```
rptd.97
rptd.98
rptd.99
rptd.00
smsa1.77
smsa1.78
smsa1.79
smsa1.80
smsa1.81
smsa1.82
smsa1.83
smsa1.84
smsa1.85
smsa1.86
smsa1.87
smsa1.88
smsa1.89
smsa1.90
smsa1.91
smsa1.92
smsa1.93
smsa1.94
smsa1.95
smsa1.96
smsa1.97
smsa1.98
smsa1.99
smsa1.00
cnty1.77
cnty1.78
cnty1.79
cnty1.80
cnty1.81
cnty1.82
cnty1.83
cnty1.84
cnty1.85
cnty1.86
cnty1.87
cnty1.88
cnty1.89
cnty1.90
cnty1.91
cnty1.92
cnty1.93
cnty1.94
cnty1.95
cnty1.96
cnty1.97
cnty1.98
cnty1.99
cnty1.00
grp.77
grp.78
grp.79
grp.80
grp.81
```

```
grp.82
grp.83
grp.84
grp.85
grp.86
grp.87
grp.88
grp.89
grp.90
grp.91
grp.92
grp.93
grp.94
grp.95
grp.96
grp.97
grp.98
grp.99
grp.00
rptmo.77
rptmo.78
rptmo.79
rptmo.80
rptmo.81
rptmo.82
rptmo.83
rptmo.84
rptmo.85
rptmo.86
rptmo.87
rptmo.88
rptmo.89
rptmo.90
rptmo.91
rptmo.92
rptmo.93
rptmo.94
rptmo.95
rptmo.96
rptmo.97
rptmo.98
rptmo.99
rptmo.00.
execute.
SAVE TRANSLATE OUTFILE='C:\WINDOWS\Desktop\XX_MU1.xls'
 /TYPE=XLS /MAP /REPLACE /FIELDNAMES / KEEP
ori_code
mu.77.01
mu.77.02
mu.77.03
mu.77.04
mu.77.05
mu.77.06
mu.77.07
mu.77.08
mu.77.09
mu.77.10
```

mu.77.11
mu.77.12
mu.78.01
mu.78.02
mu.78.03
mu.78.04
mu.78.05
mu.78.06
mu.78.07
mu.78.08
mu.78.09
mu.78.10
mu.78.11
mu.78.12
mu.79.01
mu.79.02
mu.79.03
mu.79.04
mu.79.05
mu.79.06
mu.79.07
mu.79.08
mu.79.09
mu.79.10
mu.79.11
mu.79.12
mu.80.01
mu.80.02
mu.80.03
mu.80.04
mu.80.05
mu.80.06
mu.80.07
mu.80.08
mu.80.09
mu.80.10
mu.80.11
mu.80.12
mu.81.01
mu.81.02
mu.81.03
mu.81.04
mu.81.05
mu.81.06
mu.81.07
mu.81.08
mu.81.09
mu.81.10
mu.81.11
mu.81.12
mu.82.01
mu.82.02
mu.82.03
mu.82.04
mu.82.05
mu.82.06
mu.82.07

mu.82.08
mu.82.09
mu.82.10
mu.82.11
mu.82.12
mu.83.01
mu.83.02
mu.83.03
mu.83.04
mu.83.05
mu.83.06
mu.83.07
mu.83.08
mu.83.09
mu.83.10
mu.83.11
mu.83.12
mu.84.01
mu.84.02
mu.84.03
mu.84.04
mu.84.05
mu.84.06
mu.84.07
mu.84.08
mu.84.09
mu.84.10
mu.84.11
mu.84.12
mu.85.01
mu.85.02
mu.85.03
mu.85.04
mu.85.05
mu.85.06
mu.85.07
mu.85.08
mu.85.09
mu.85.10
mu.85.11
mu.85.12
mu.86.01
mu.86.02
mu.86.03
mu.86.04
mu.86.05
mu.86.06
mu.86.07
mu.86.08
mu.86.09
mu.86.10
mu.86.11
mu.86.12
mu.87.01
mu.87.02
mu.87.03
mu.87.04

```
mu.87.05
mu.87.06
mu.87.07
mu.87.08
mu.87.09
mu.87.10
mu.87.11
mu.87.12
mu.88.01
mu.88.02
mu.88.03
mu.88.04
mu.88.05
mu.88.06
mu.88.07
mu.88.08
mu.88.09
mu.88.10
mu.88.11
mu.88.12.
execute.
SAVE TRANSLATE OUTFILE='C:\WINDOWS\Desktop\XX_MU2.xls'
 /TYPE=XLS /MAP /REPLACE /FIELDNAMES / KEEP
ori_code
mu.89.01
mu.89.02
mu.89.03
mu.89.04
mu.89.05
mu.89.06
mu.89.07
mu.89.08
mu.89.09
mu.89.10
mu.89.11
mu.89.12
mu.90.01
mu.90.02
mu.90.03
mu.90.04
mu.90.05
mu.90.06
mu.90.07
mu.90.08
mu.90.09
mu.90.10
mu.90.11
mu.90.12
mu.91.01
mu.91.02
mu.91.03
mu.91.04
mu.91.05
mu.91.06
mu.91.07
mu.91.08
mu.91.09
```

mu.91.10
mu.91.11
mu.91.12
mu.92.01
mu.92.02
mu.92.03
mu.92.04
mu.92.05
mu.92.06
mu.92.07
mu.92.08
mu.92.09
mu.92.10
mu.92.11
mu.92.12
mu.93.01
mu.93.02
mu.93.03
mu.93.04
mu.93.05
mu.93.06
mu.93.07
mu.93.08
mu.93.09
mu.93.10
mu.93.11
mu.93.12
mu.94.01
mu.94.02
mu.94.03
mu.94.04
mu.94.05
mu.94.06
mu.94.07
mu.94.08
mu.94.09
mu.94.10
mu.94.11
mu.94.12
mu.95.01
mu.95.02
mu.95.03
mu.95.04
mu.95.05
mu.95.06
mu.95.07
mu.95.08
mu.95.09
mu.95.10
mu.95.11
mu.95.12
mu.96.01
mu.96.02
mu.96.03
mu.96.04
mu.96.05
mu.96.06

```
mu.96.07
mu.96.08
mu.96.09
mu.96.10
mu.96.11
mu.96.12
mu.97.01
mu.97.02
mu.97.03
mu.97.04
mu.97.05
mu.97.06
mu.97.07
mu.97.08
mu.97.09
mu.97.10
mu.97.11
mu.97.12
mu.98.01
mu.98.02
mu.98.03
mu.98.04
mu.98.05
mu.98.06
mu.98.07
mu.98.08
mu.98.09
mu.98.10
mu.98.11
mu.98.12
mu.99.01
mu.99.02
mu.99.03
mu.99.04
mu.99.05
mu.99.06
mu.99.07
mu.99.08
mu.99.09
mu.99.10
mu.99.11
mu.99.12
mu.00.01
mu.00.02
mu.00.03
mu.00.04
mu.00.05
mu.00.06
mu.00.07
mu.00.08
mu.00.09
mu.00.10
mu.00.11
mu.00.12.
execute.
SAVE TRANSLATE OUTFILE='C:\WINDOWS\Desktop\XX_RA1.xls'
 /TYPE=XLS /MAP /REPLACE /FIELDNAMES / KEEP
```

ori_code
ra.77.01
ra.77.02
ra.77.03
ra.77.04
ra.77.05
ra.77.06
ra.77.07
ra.77.08
ra.77.09
ra.77.10
ra.77.11
ra.77.12
ra.78.01
ra.78.02
ra.78.03
ra.78.04
ra.78.05
ra.78.06
ra.78.07
ra.78.08
ra.78.09
ra.78.10
ra.78.11
ra.78.12
ra.79.01
ra.79.02
ra.79.03
ra.79.04
ra.79.05
ra.79.06
ra.79.07
ra.79.08
ra.79.09
ra.79.10
ra.79.11
ra.79.12
ra.80.01
ra.80.02
ra.80.03
ra.80.04
ra.80.05
ra.80.06
ra.80.07
ra.80.08
ra.80.09
ra.80.10
ra.80.11
ra.80.12
ra.81.01
ra.81.02
ra.81.03
ra.81.04
ra.81.05
ra.81.06
ra.81.07
ra.81.08

ra.81.09
ra.81.10
ra.81.11
ra.81.12
ra.82.01
ra.82.02
ra.82.03
ra.82.04
ra.82.05
ra.82.06
ra.82.07
ra.82.08
ra.82.09
ra.82.10
ra.82.11
ra.82.12
ra.83.01
ra.83.02
ra.83.03
ra.83.04
ra.83.05
ra.83.06
ra.83.07
ra.83.08
ra.83.09
ra.83.10
ra.83.11
ra.83.12
ra.84.01
ra.84.02
ra.84.03
ra.84.04
ra.84.05
ra.84.06
ra.84.07
ra.84.08
ra.84.09
ra.84.10
ra.84.11
ra.84.12
ra.85.01
ra.85.02
ra.85.03
ra.85.04
ra.85.05
ra.85.06
ra.85.07
ra.85.08
ra.85.09
ra.85.10
ra.85.11
ra.85.12
ra.86.01
ra.86.02
ra.86.03
ra.86.04
ra.86.05

```
ra.86.06
ra.86.07
ra.86.08
ra.86.09
ra.86.10
ra.86.11
ra.86.12
ra.87.01
ra.87.02
ra.87.03
ra.87.04
ra.87.05
ra.87.06
ra.87.07
ra.87.08
ra.87.09
ra.87.10
ra.87.11
ra.87.12
ra.88.01
ra.88.02
ra.88.03
ra.88.04
ra.88.05
ra.88.06
ra.88.07
ra.88.08
ra.88.09
ra.88.10
ra.88.11
ra.88.12    .
execute.
SAVE TRANSLATE OUTFILE='C:\WINDOWS\Desktop\XX_RA2.xls'
 /TYPE=XLS /MAP /REPLACE /FIELDNAMES / KEEP
ori_code
ra.89.01
ra.89.02
ra.89.03
ra.89.04
ra.89.05
ra.89.06
ra.89.07
ra.89.08
ra.89.09
ra.89.10
ra.89.11
ra.89.12
ra.90.01
ra.90.02
ra.90.03
ra.90.04
ra.90.05
ra.90.06
ra.90.07
ra.90.08
ra.90.09
ra.90.10
```

ra.90.11
ra.90.12
ra.91.01
ra.91.02
ra.91.03
ra.91.04
ra.91.05
ra.91.06
ra.91.07
ra.91.08
ra.91.09
ra.91.10
ra.91.11
ra.91.12
ra.92.01
ra.92.02
ra.92.03
ra.92.04
ra.92.05
ra.92.06
ra.92.07
ra.92.08
ra.92.09
ra.92.10
ra.92.11
ra.92.12
ra.93.01
ra.93.02
ra.93.03
ra.93.04
ra.93.05
ra.93.06
ra.93.07
ra.93.08
ra.93.09
ra.93.10
ra.93.11
ra.93.12
ra.94.01
ra.94.02
ra.94.03
ra.94.04
ra.94.05
ra.94.06
ra.94.07
ra.94.08
ra.94.09
ra.94.10
ra.94.11
ra.94.12
ra.95.01
ra.95.02
ra.95.03
ra.95.04
ra.95.05
ra.95.06
ra.95.07

```
ra.95.08
ra.95.09
ra.95.10
ra.95.11
ra.95.12
ra.96.01
ra.96.02
ra.96.03
ra.96.04
ra.96.05
ra.96.06
ra.96.07
ra.96.08
ra.96.09
ra.96.10
ra.96.11
ra.96.12
ra.97.01
ra.97.02
ra.97.03
ra.97.04
ra.97.05
ra.97.06
ra.97.07
ra.97.08
ra.97.09
ra.97.10
ra.97.11
ra.97.12
ra.98.01
ra.98.02
ra.98.03
ra.98.04
ra.98.05
ra.98.06
ra.98.07
ra.98.08
ra.98.09
ra.98.10
ra.98.11
ra.98.12
ra.99.01
ra.99.02
ra.99.03
ra.99.04
ra.99.05
ra.99.06
ra.99.07
ra.99.08
ra.99.09
ra.99.10
ra.99.11
ra.99.12
ra.00.01
ra.00.02
ra.00.03
ra.00.04
```

```
ra.00.05
ra.00.06
ra.00.07
ra.00.08
ra.00.09
ra.00.10
ra.00.11
ra.00.12    .
execute.
SAVE TRANSLATE OUTFILE='C:\WINDOWS\Desktop\XX_RO1.xls'
 /TYPE=XLS /MAP /REPLACE /FIELDNAMES / KEEP
ori_code
ro.77.01
ro.77.02
ro.77.03
ro.77.04
ro.77.05
ro.77.06
ro.77.07
ro.77.08
ro.77.09
ro.77.10
ro.77.11
ro.77.12
ro.78.01
ro.78.02
ro.78.03
ro.78.04
ro.78.05
ro.78.06
ro.78.07
ro.78.08
ro.78.09
ro.78.10
ro.78.11
ro.78.12
ro.79.01
ro.79.02
ro.79.03
ro.79.04
ro.79.05
ro.79.06
ro.79.07
ro.79.08
ro.79.09
ro.79.10
ro.79.11
ro.79.12
ro.80.01
ro.80.02
ro.80.03
ro.80.04
ro.80.05
ro.80.06
ro.80.07
ro.80.08
ro.80.09
```

```
ro.80.10
ro.80.11
ro.80.12
ro.81.01
ro.81.02
ro.81.03
ro.81.04
ro.81.05
ro.81.06
ro.81.07
ro.81.08
ro.81.09
ro.81.10
ro.81.11
ro.81.12
ro.82.01
ro.82.02
ro.82.03
ro.82.04
ro.82.05
ro.82.06
ro.82.07
ro.82.08
ro.82.09
ro.82.10
ro.82.11
ro.82.12
ro.83.01
ro.83.02
ro.83.03
ro.83.04
ro.83.05
ro.83.06
ro.83.07
ro.83.08
ro.83.09
ro.83.10
ro.83.11
ro.83.12
ro.84.01
ro.84.02
ro.84.03
ro.84.04
ro.84.05
ro.84.06
ro.84.07
ro.84.08
ro.84.09
ro.84.10
ro.84.11
ro.84.12
ro.85.01
ro.85.02
ro.85.03
ro.85.04
ro.85.05
ro.85.06
```

```
ro.85.07
ro.85.08
ro.85.09
ro.85.10
ro.85.11
ro.85.12
ro.86.01
ro.86.02
ro.86.03
ro.86.04
ro.86.05
ro.86.06
ro.86.07
ro.86.08
ro.86.09
ro.86.10
ro.86.11
ro.86.12
ro.87.01
ro.87.02
ro.87.03
ro.87.04
ro.87.05
ro.87.06
ro.87.07
ro.87.08
ro.87.09
ro.87.10
ro.87.11
ro.87.12
ro.88.01
ro.88.02
ro.88.03
ro.88.04
ro.88.05
ro.88.06
ro.88.07
ro.88.08
ro.88.09
ro.88.10
ro.88.11
ro.88.12    .
execute.
SAVE TRANSLATE OUTFILE='C:\WINDOWS\Desktop\XX_RO2.xls'
 /TYPE=XLS /MAP /REPLACE /FIELDNAMES / KEEP
ori_code
ro.89.01
ro.89.02
ro.89.03
ro.89.04
ro.89.05
ro.89.06
ro.89.07
ro.89.08
ro.89.09
ro.89.10
ro.89.11
```

ro.89.12
ro.90.01
ro.90.02
ro.90.03
ro.90.04
ro.90.05
ro.90.06
ro.90.07
ro.90.08
ro.90.09
ro.90.10
ro.90.11
ro.90.12
ro.91.01
ro.91.02
ro.91.03
ro.91.04
ro.91.05
ro.91.06
ro.91.07
ro.91.08
ro.91.09
ro.91.10
ro.91.11
ro.91.12
ro.92.01
ro.92.02
ro.92.03
ro.92.04
ro.92.05
ro.92.06
ro.92.07
ro.92.08
ro.92.09
ro.92.10
ro.92.11
ro.92.12
ro.93.01
ro.93.02
ro.93.03
ro.93.04
ro.93.05
ro.93.06
ro.93.07
ro.93.08
ro.93.09
ro.93.10
ro.93.11
ro.93.12
ro.94.01
ro.94.02
ro.94.03
ro.94.04
ro.94.05
ro.94.06
ro.94.07
ro.94.08

ro.94.09
ro.94.10
ro.94.11
ro.94.12
ro.95.01
ro.95.02
ro.95.03
ro.95.04
ro.95.05
ro.95.06
ro.95.07
ro.95.08
ro.95.09
ro.95.10
ro.95.11
ro.95.12
ro.96.01
ro.96.02
ro.96.03
ro.96.04
ro.96.05
ro.96.06
ro.96.07
ro.96.08
ro.96.09
ro.96.10
ro.96.11
ro.96.12
ro.97.01
ro.97.02
ro.97.03
ro.97.04
ro.97.05
ro.97.06
ro.97.07
ro.97.08
ro.97.09
ro.97.10
ro.97.11
ro.97.12
ro.98.01
ro.98.02
ro.98.03
ro.98.04
ro.98.05
ro.98.06
ro.98.07
ro.98.08
ro.98.09
ro.98.10
ro.98.11
ro.98.12
ro.99.01
ro.99.02
ro.99.03
ro.99.04
ro.99.05

```
ro.99.06
ro.99.07
ro.99.08
ro.99.09
ro.99.10
ro.99.11
ro.99.12
ro.00.01
ro.00.02
ro.00.03
ro.00.04
ro.00.05
ro.00.06
ro.00.07
ro.00.08
ro.00.09
ro.00.10
ro.00.11
ro.00.12    .
execute.
SAVE TRANSLATE OUTFILE='C:\WINDOWS\Desktop\XX_AS1.xls'
 /TYPE=XLS /MAP /REPLACE /FIELDNAMES / KEEP
ori_code
as.77.01
as.77.02
as.77.03
as.77.04
as.77.05
as.77.06
as.77.07
as.77.08
as.77.09
as.77.10
as.77.11
as.77.12
as.78.01
as.78.02
as.78.03
as.78.04
as.78.05
as.78.06
as.78.07
as.78.08
as.78.09
as.78.10
as.78.11
as.78.12
as.79.01
as.79.02
as.79.03
as.79.04
as.79.05
as.79.06
as.79.07
as.79.08
as.79.09
as.79.10
```

as.79.11
as.79.12
as.80.01
as.80.02
as.80.03
as.80.04
as.80.05
as.80.06
as.80.07
as.80.08
as.80.09
as.80.10
as.80.11
as.80.12
as.81.01
as.81.02
as.81.03
as.81.04
as.81.05
as.81.06
as.81.07
as.81.08
as.81.09
as.81.10
as.81.11
as.81.12
as.82.01
as.82.02
as.82.03
as.82.04
as.82.05
as.82.06
as.82.07
as.82.08
as.82.09
as.82.10
as.82.11
as.82.12
as.83.01
as.83.02
as.83.03
as.83.04
as.83.05
as.83.06
as.83.07
as.83.08
as.83.09
as.83.10
as.83.11
as.83.12
as.84.01
as.84.02
as.84.03
as.84.04
as.84.05
as.84.06
as.84.07

```
as.84.08
as.84.09
as.84.10
as.84.11
as.84.12
as.85.01
as.85.02
as.85.03
as.85.04
as.85.05
as.85.06
as.85.07
as.85.08
as.85.09
as.85.10
as.85.11
as.85.12
as.86.01
as.86.02
as.86.03
as.86.04
as.86.05
as.86.06
as.86.07
as.86.08
as.86.09
as.86.10
as.86.11
as.86.12
as.87.01
as.87.02
as.87.03
as.87.04
as.87.05
as.87.06
as.87.07
as.87.08
as.87.09
as.87.10
as.87.11
as.87.12
as.88.01
as.88.02
as.88.03
as.88.04
as.88.05
as.88.06
as.88.07
as.88.08
as.88.09
as.88.10
as.88.11
as.88.12.
execute.
SAVE TRANSLATE OUTFILE='C:\WINDOWS\Desktop\XX_AS2.xls'
 /TYPE=XLS /MAP /REPLACE /FIELDNAMES / KEEP
ori_code
```

as.89.01
as.89.02
as.89.03
as.89.04
as.89.05
as.89.06
as.89.07
as.89.08
as.89.09
as.89.10
as.89.11
as.89.12
as.90.01
as.90.02
as.90.03
as.90.04
as.90.05
as.90.06
as.90.07
as.90.08
as.90.09
as.90.10
as.90.11
as.90.12
as.91.01
as.91.02
as.91.03
as.91.04
as.91.05
as.91.06
as.91.07
as.91.08
as.91.09
as.91.10
as.91.11
as.91.12
as.92.01
as.92.02
as.92.03
as.92.04
as.92.05
as.92.06
as.92.07
as.92.08
as.92.09
as.92.10
as.92.11
as.92.12
as.93.01
as.93.02
as.93.03
as.93.04
as.93.05
as.93.06
as.93.07
as.93.08
as.93.09

as.93.10
as.93.11
as.93.12
as.94.01
as.94.02
as.94.03
as.94.04
as.94.05
as.94.06
as.94.07
as.94.08
as.94.09
as.94.10
as.94.11
as.94.12
as.95.01
as.95.02
as.95.03
as.95.04
as.95.05
as.95.06
as.95.07
as.95.08
as.95.09
as.95.10
as.95.11
as.95.12
as.96.01
as.96.02
as.96.03
as.96.04
as.96.05
as.96.06
as.96.07
as.96.08
as.96.09
as.96.10
as.96.11
as.96.12
as.97.01
as.97.02
as.97.03
as.97.04
as.97.05
as.97.06
as.97.07
as.97.08
as.97.09
as.97.10
as.97.11
as.97.12
as.98.01
as.98.02
as.98.03
as.98.04
as.98.05
as.98.06

```
as.98.07
as.98.08
as.98.09
as.98.10
as.98.11
as.98.12
as.99.01
as.99.02
as.99.03
as.99.04
as.99.05
as.99.06
as.99.07
as.99.08
as.99.09
as.99.10
as.99.11
as.99.12
as.00.01
as.00.02
as.00.03
as.00.04
as.00.05
as.00.06
as.00.07
as.00.08
as.00.09
as.00.10
as.00.11
as.00.12.
execute.
SAVE TRANSLATE OUTFILE='C:\WINDOWS\Desktop\XX_BU1.xls'
 /TYPE=XLS /MAP /REPLACE /FIELDNAMES / KEEP
ori_code
bu.77.01
bu.77.02
bu.77.03
bu.77.04
bu.77.05
bu.77.06
bu.77.07
bu.77.08
bu.77.09
bu.77.10
bu.77.11
bu.77.12
bu.78.01
bu.78.02
bu.78.03
bu.78.04
bu.78.05
bu.78.06
bu.78.07
bu.78.08
bu.78.09
bu.78.10
bu.78.11
```

bu.78.12
bu.79.01
bu.79.02
bu.79.03
bu.79.04
bu.79.05
bu.79.06
bu.79.07
bu.79.08
bu.79.09
bu.79.10
bu.79.11
bu.79.12
bu.80.01
bu.80.02
bu.80.03
bu.80.04
bu.80.05
bu.80.06
bu.80.07
bu.80.08
bu.80.09
bu.80.10
bu.80.11
bu.80.12
bu.81.01
bu.81.02
bu.81.03
bu.81.04
bu.81.05
bu.81.06
bu.81.07
bu.81.08
bu.81.09
bu.81.10
bu.81.11
bu.81.12
bu.82.01
bu.82.02
bu.82.03
bu.82.04
bu.82.05
bu.82.06
bu.82.07
bu.82.08
bu.82.09
bu.82.10
bu.82.11
bu.82.12
bu.83.01
bu.83.02
bu.83.03
bu.83.04
bu.83.05
bu.83.06
bu.83.07
bu.83.08

bu.83.09
bu.83.10
bu.83.11
bu.83.12
bu.84.01
bu.84.02
bu.84.03
bu.84.04
bu.84.05
bu.84.06
bu.84.07
bu.84.08
bu.84.09
bu.84.10
bu.84.11
bu.84.12
bu.85.01
bu.85.02
bu.85.03
bu.85.04
bu.85.05
bu.85.06
bu.85.07
bu.85.08
bu.85.09
bu.85.10
bu.85.11
bu.85.12
bu.86.01
bu.86.02
bu.86.03
bu.86.04
bu.86.05
bu.86.06
bu.86.07
bu.86.08
bu.86.09
bu.86.10
bu.86.11
bu.86.12
bu.87.01
bu.87.02
bu.87.03
bu.87.04
bu.87.05
bu.87.06
bu.87.07
bu.87.08
bu.87.09
bu.87.10
bu.87.11
bu.87.12
bu.88.01
bu.88.02
bu.88.03
bu.88.04
bu.88.05

```
bu.88.06
bu.88.07
bu.88.08
bu.88.09
bu.88.10
bu.88.11
bu.88.12.
execute.
SAVE TRANSLATE OUTFILE='C:\WINDOWS\Desktop\XX_BU2.xls'
 /TYPE=XLS /MAP /REPLACE /FIELDNAMES / KEEP
ori_code
bu.89.01
bu.89.02
bu.89.03
bu.89.04
bu.89.05
bu.89.06
bu.89.07
bu.89.08
bu.89.09
bu.89.10
bu.89.11
bu.89.12
bu.90.01
bu.90.02
bu.90.03
bu.90.04
bu.90.05
bu.90.06
bu.90.07
bu.90.08
bu.90.09
bu.90.10
bu.90.11
bu.90.12
bu.91.01
bu.91.02
bu.91.03
bu.91.04
bu.91.05
bu.91.06
bu.91.07
bu.91.08
bu.91.09
bu.91.10
bu.91.11
bu.91.12
bu.92.01
bu.92.02
bu.92.03
bu.92.04
bu.92.05
bu.92.06
bu.92.07
bu.92.08
bu.92.09
bu.92.10
```

bu.92.11
bu.92.12
bu.93.01
bu.93.02
bu.93.03
bu.93.04
bu.93.05
bu.93.06
bu.93.07
bu.93.08
bu.93.09
bu.93.10
bu.93.11
bu.93.12
bu.94.01
bu.94.02
bu.94.03
bu.94.04
bu.94.05
bu.94.06
bu.94.07
bu.94.08
bu.94.09
bu.94.10
bu.94.11
bu.94.12
bu.95.01
bu.95.02
bu.95.03
bu.95.04
bu.95.05
bu.95.06
bu.95.07
bu.95.08
bu.95.09
bu.95.10
bu.95.11
bu.95.12
bu.96.01
bu.96.02
bu.96.03
bu.96.04
bu.96.05
bu.96.06
bu.96.07
bu.96.08
bu.96.09
bu.96.10
bu.96.11
bu.96.12
bu.97.01
bu.97.02
bu.97.03
bu.97.04
bu.97.05
bu.97.06
bu.97.07

```
bu.97.08
bu.97.09
bu.97.10
bu.97.11
bu.97.12
bu.98.01
bu.98.02
bu.98.03
bu.98.04
bu.98.05
bu.98.06
bu.98.07
bu.98.08
bu.98.09
bu.98.10
bu.98.11
bu.98.12
bu.99.01
bu.99.02
bu.99.03
bu.99.04
bu.99.05
bu.99.06
bu.99.07
bu.99.08
bu.99.09
bu.99.10
bu.99.11
bu.99.12
bu.00.01
bu.00.02
bu.00.03
bu.00.04
bu.00.05
bu.00.06
bu.00.07
bu.00.08
bu.00.09
bu.00.10
bu.00.11
bu.00.12.
execute.
SAVE TRANSLATE OUTFILE='C:\WINDOWS\Desktop\XX_LA1.xls'
 /TYPE=XLS /MAP /REPLACE /FIELDNAMES / KEEP
ori_code
la.77.01
la.77.02
la.77.03
la.77.04
la.77.05
la.77.06
la.77.07
la.77.08
la.77.09
la.77.10
la.77.11
la.77.12
```

la.78.01
la.78.02
la.78.03
la.78.04
la.78.05
la.78.06
la.78.07
la.78.08
la.78.09
la.78.10
la.78.11
la.78.12
la.79.01
la.79.02
la.79.03
la.79.04
la.79.05
la.79.06
la.79.07
la.79.08
la.79.09
la.79.10
la.79.11
la.79.12
la.80.01
la.80.02
la.80.03
la.80.04
la.80.05
la.80.06
la.80.07
la.80.08
la.80.09
la.80.10
la.80.11
la.80.12
la.81.01
la.81.02
la.81.03
la.81.04
la.81.05
la.81.06
la.81.07
la.81.08
la.81.09
la.81.10
la.81.11
la.81.12
la.82.01
la.82.02
la.82.03
la.82.04
la.82.05
la.82.06
la.82.07
la.82.08
la.82.09

la.82.10
la.82.11
la.82.12
la.83.01
la.83.02
la.83.03
la.83.04
la.83.05
la.83.06
la.83.07
la.83.08
la.83.09
la.83.10
la.83.11
la.83.12
la.84.01
la.84.02
la.84.03
la.84.04
la.84.05
la.84.06
la.84.07
la.84.08
la.84.09
la.84.10
la.84.11
la.84.12
la.85.01
la.85.02
la.85.03
la.85.04
la.85.05
la.85.06
la.85.07
la.85.08
la.85.09
la.85.10
la.85.11
la.85.12
la.86.01
la.86.02
la.86.03
la.86.04
la.86.05
la.86.06
la.86.07
la.86.08
la.86.09
la.86.10
la.86.11
la.86.12
la.87.01
la.87.02
la.87.03
la.87.04
la.87.05
la.87.06

```
la.87.07
la.87.08
la.87.09
la.87.10
la.87.11
la.87.12
la.88.01
la.88.02
la.88.03
la.88.04
la.88.05
la.88.06
la.88.07
la.88.08
la.88.09
la.88.10
la.88.11
la.88.12    .
execute.
SAVE TRANSLATE OUTFILE='C:\WINDOWS\Desktop\XX_LA2.xls'
 /TYPE=XLS /MAP /REPLACE /FIELDNAMES / KEEP
ori_code
la.89.01
la.89.02
la.89.03
la.89.04
la.89.05
la.89.06
la.89.07
la.89.08
la.89.09
la.89.10
la.89.11
la.89.12
la.90.01
la.90.02
la.90.03
la.90.04
la.90.05
la.90.06
la.90.07
la.90.08
la.90.09
la.90.10
la.90.11
la.90.12
la.91.01
la.91.02
la.91.03
la.91.04
la.91.05
la.91.06
la.91.07
la.91.08
la.91.09
la.91.10
la.91.11
```

```
la.91.12
la.92.01
la.92.02
la.92.03
la.92.04
la.92.05
la.92.06
la.92.07
la.92.08
la.92.09
la.92.10
la.92.11
la.92.12
la.93.01
la.93.02
la.93.03
la.93.04
la.93.05
la.93.06
la.93.07
la.93.08
la.93.09
la.93.10
la.93.11
la.93.12
la.94.01
la.94.02
la.94.03
la.94.04
la.94.05
la.94.06
la.94.07
la.94.08
la.94.09
la.94.10
la.94.11
la.94.12
la.95.01
la.95.02
la.95.03
la.95.04
la.95.05
la.95.06
la.95.07
la.95.08
la.95.09
la.95.10
la.95.11
la.95.12
la.96.01
la.96.02
la.96.03
la.96.04
la.96.05
la.96.06
la.96.07
la.96.08
```

```
la.96.09
la.96.10
la.96.11
la.96.12
la.97.01
la.97.02
la.97.03
la.97.04
la.97.05
la.97.06
la.97.07
la.97.08
la.97.09
la.97.10
la.97.11
la.97.12
la.98.01
la.98.02
la.98.03
la.98.04
la.98.05
la.98.06
la.98.07
la.98.08
la.98.09
la.98.10
la.98.11
la.98.12
la.99.01
la.99.02
la.99.03
la.99.04
la.99.05
la.99.06
la.99.07
la.99.08
la.99.09
la.99.10
la.99.11
la.99.12
la.00.01
la.00.02
la.00.03
la.00.04
la.00.05
la.00.06
la.00.07
la.00.08
la.00.09
la.00.10
la.00.11
la.00.12    .
execute.
SAVE TRANSLATE OUTFILE='C:\WINDOWS\Desktop\XX_MV1.xls'
 /TYPE=XLS /MAP /REPLACE /FIELDNAMES / KEEP
ori_code
mv.77.01
```

mv.77.02
mv.77.03
mv.77.04
mv.77.05
mv.77.06
mv.77.07
mv.77.08
mv.77.09
mv.77.10
mv.77.11
mv.77.12
mv.78.01
mv.78.02
mv.78.03
mv.78.04
mv.78.05
mv.78.06
mv.78.07
mv.78.08
mv.78.09
mv.78.10
mv.78.11
mv.78.12
mv.79.01
mv.79.02
mv.79.03
mv.79.04
mv.79.05
mv.79.06
mv.79.07
mv.79.08
mv.79.09
mv.79.10
mv.79.11
mv.79.12
mv.80.01
mv.80.02
mv.80.03
mv.80.04
mv.80.05
mv.80.06
mv.80.07
mv.80.08
mv.80.09
mv.80.10
mv.80.11
mv.80.12
mv.81.01
mv.81.02
mv.81.03
mv.81.04
mv.81.05
mv.81.06
mv.81.07
mv.81.08
mv.81.09
mv.81.10

```
mv.81.11
mv.81.12
mv.82.01
mv.82.02
mv.82.03
mv.82.04
mv.82.05
mv.82.06
mv.82.07
mv.82.08
mv.82.09
mv.82.10
mv.82.11
mv.82.12
mv.83.01
mv.83.02
mv.83.03
mv.83.04
mv.83.05
mv.83.06
mv.83.07
mv.83.08
mv.83.09
mv.83.10
mv.83.11
mv.83.12
mv.84.01
mv.84.02
mv.84.03
mv.84.04
mv.84.05
mv.84.06
mv.84.07
mv.84.08
mv.84.09
mv.84.10
mv.84.11
mv.84.12
mv.85.01
mv.85.02
mv.85.03
mv.85.04
mv.85.05
mv.85.06
mv.85.07
mv.85.08
mv.85.09
mv.85.10
mv.85.11
mv.85.12
mv.86.01
mv.86.02
mv.86.03
mv.86.04
mv.86.05
mv.86.06
mv.86.07
```

```
mv.86.08
mv.86.09
mv.86.10
mv.86.11
mv.86.12
mv.87.01
mv.87.02
mv.87.03
mv.87.04
mv.87.05
mv.87.06
mv.87.07
mv.87.08
mv.87.09
mv.87.10
mv.87.11
mv.87.12
mv.88.01
mv.88.02
mv.88.03
mv.88.04
mv.88.05
mv.88.06
mv.88.07
mv.88.08
mv.88.09
mv.88.10
mv.88.11
mv.88.12.
execute.
SAVE TRANSLATE OUTFILE='C:\WINDOWS\Desktop\XX_MV2.xls'
 /TYPE=XLS /MAP /REPLACE /FIELDNAMES / KEEP
ori_code
mv.89.01
mv.89.02
mv.89.03
mv.89.04
mv.89.05
mv.89.06
mv.89.07
mv.89.08
mv.89.09
mv.89.10
mv.89.11
mv.89.12
mv.90.01
mv.90.02
mv.90.03
mv.90.04
mv.90.05
mv.90.06
mv.90.07
mv.90.08
mv.90.09
mv.90.10
mv.90.11
mv.90.12
```

mv.91.01
mv.91.02
mv.91.03
mv.91.04
mv.91.05
mv.91.06
mv.91.07
mv.91.08
mv.91.09
mv.91.10
mv.91.11
mv.91.12
mv.92.01
mv.92.02
mv.92.03
mv.92.04
mv.92.05
mv.92.06
mv.92.07
mv.92.08
mv.92.09
mv.92.10
mv.92.11
mv.92.12
mv.93.01
mv.93.02
mv.93.03
mv.93.04
mv.93.05
mv.93.06
mv.93.07
mv.93.08
mv.93.09
mv.93.10
mv.93.11
mv.93.12
mv.94.01
mv.94.02
mv.94.03
mv.94.04
mv.94.05
mv.94.06
mv.94.07
mv.94.08
mv.94.09
mv.94.10
mv.94.11
mv.94.12
mv.95.01
mv.95.02
mv.95.03
mv.95.04
mv.95.05
mv.95.06
mv.95.07
mv.95.08
mv.95.09

```
mv.95.10
mv.95.11
mv.95.12
mv.96.01
mv.96.02
mv.96.03
mv.96.04
mv.96.05
mv.96.06
mv.96.07
mv.96.08
mv.96.09
mv.96.10
mv.96.11
mv.96.12
mv.97.01
mv.97.02
mv.97.03
mv.97.04
mv.97.05
mv.97.06
mv.97.07
mv.97.08
mv.97.09
mv.97.10
mv.97.11
mv.97.12
mv.98.01
mv.98.02
mv.98.03
mv.98.04
mv.98.05
mv.98.06
mv.98.07
mv.98.08
mv.98.09
mv.98.10
mv.98.11
mv.98.12
mv.99.01
mv.99.02
mv.99.03
mv.99.04
mv.99.05
mv.99.06
mv.99.07
mv.99.08
mv.99.09
mv.99.10
mv.99.11
mv.99.12
mv.00.01
mv.00.02
mv.00.03
mv.00.04
mv.00.05
mv.00.06
```

```
mv.00.07
mv.00.08
mv.00.09
mv.00.10
mv.00.11
mv.00.12.
execute.
SAVE TRANSLATE OUTFILE='C:\WINDOWS\Desktop\XX_CI1.xls'
 /TYPE=XLS /MAP /REPLACE /FIELDNAMES / KEEP
ori_code
ci.77.01
ci.77.02
ci.77.03
ci.77.04
ci.77.05
ci.77.06
ci.77.07
ci.77.08
ci.77.09
ci.77.10
ci.77.11
ci.77.12
ci.78.01
ci.78.02
ci.78.03
ci.78.04
ci.78.05
ci.78.06
ci.78.07
ci.78.08
ci.78.09
ci.78.10
ci.78.11
ci.78.12
ci.79.01
ci.79.02
ci.79.03
ci.79.04
ci.79.05
ci.79.06
ci.79.07
ci.79.08
ci.79.09
ci.79.10
ci.79.11
ci.79.12
ci.80.01
ci.80.02
ci.80.03
ci.80.04
ci.80.05
ci.80.06
ci.80.07
ci.80.08
ci.80.09
ci.80.10
ci.80.11
```

ci.80.12
ci.81.01
ci.81.02
ci.81.03
ci.81.04
ci.81.05
ci.81.06
ci.81.07
ci.81.08
ci.81.09
ci.81.10
ci.81.11
ci.81.12
ci.82.01
ci.82.02
ci.82.03
ci.82.04
ci.82.05
ci.82.06
ci.82.07
ci.82.08
ci.82.09
ci.82.10
ci.82.11
ci.82.12
ci.83.01
ci.83.02
ci.83.03
ci.83.04
ci.83.05
ci.83.06
ci.83.07
ci.83.08
ci.83.09
ci.83.10
ci.83.11
ci.83.12
ci.84.01
ci.84.02
ci.84.03
ci.84.04
ci.84.05
ci.84.06
ci.84.07
ci.84.08
ci.84.09
ci.84.10
ci.84.11
ci.84.12
ci.85.01
ci.85.02
ci.85.03
ci.85.04
ci.85.05
ci.85.06
ci.85.07
ci.85.08

```
ci.85.09
ci.85.10
ci.85.11
ci.85.12
ci.86.01
ci.86.02
ci.86.03
ci.86.04
ci.86.05
ci.86.06
ci.86.07
ci.86.08
ci.86.09
ci.86.10
ci.86.11
ci.86.12
ci.87.01
ci.87.02
ci.87.03
ci.87.04
ci.87.05
ci.87.06
ci.87.07
ci.87.08
ci.87.09
ci.87.10
ci.87.11
ci.87.12
ci.88.01
ci.88.02
ci.88.03
ci.88.04
ci.88.05
ci.88.06
ci.88.07
ci.88.08
ci.88.09
ci.88.10
ci.88.11
ci.88.12    .
execute.
SAVE TRANSLATE OUTFILE='C:\WINDOWS\Desktop\XX_CI2.xls'
 /TYPE=XLS /MAP /REPLACE /FIELDNAMES / KEEP
ori_code
ci.89.01
ci.89.02
ci.89.03
ci.89.04
ci.89.05
ci.89.06
ci.89.07
ci.89.08
ci.89.09
ci.89.10
ci.89.11
ci.89.12
ci.90.01
```

ci.90.02
ci.90.03
ci.90.04
ci.90.05
ci.90.06
ci.90.07
ci.90.08
ci.90.09
ci.90.10
ci.90.11
ci.90.12
ci.91.01
ci.91.02
ci.91.03
ci.91.04
ci.91.05
ci.91.06
ci.91.07
ci.91.08
ci.91.09
ci.91.10
ci.91.11
ci.91.12
ci.92.01
ci.92.02
ci.92.03
ci.92.04
ci.92.05
ci.92.06
ci.92.07
ci.92.08
ci.92.09
ci.92.10
ci.92.11
ci.92.12
ci.93.01
ci.93.02
ci.93.03
ci.93.04
ci.93.05
ci.93.06
ci.93.07
ci.93.08
ci.93.09
ci.93.10
ci.93.11
ci.93.12
ci.94.01
ci.94.02
ci.94.03
ci.94.04
ci.94.05
ci.94.06
ci.94.07
ci.94.08
ci.94.09
ci.94.10

ci.94.11
ci.94.12
ci.95.01
ci.95.02
ci.95.03
ci.95.04
ci.95.05
ci.95.06
ci.95.07
ci.95.08
ci.95.09
ci.95.10
ci.95.11
ci.95.12
ci.96.01
ci.96.02
ci.96.03
ci.96.04
ci.96.05
ci.96.06
ci.96.07
ci.96.08
ci.96.09
ci.96.10
ci.96.11
ci.96.12
ci.97.01
ci.97.02
ci.97.03
ci.97.04
ci.97.05
ci.97.06
ci.97.07
ci.97.08
ci.97.09
ci.97.10
ci.97.11
ci.97.12
ci.98.01
ci.98.02
ci.98.03
ci.98.04
ci.98.05
ci.98.06
ci.98.07
ci.98.08
ci.98.09
ci.98.10
ci.98.11
ci.98.12
ci.99.01
ci.99.02
ci.99.03
ci.99.04
ci.99.05
ci.99.06
ci.99.07

```
ci.99.08
ci.99.09
ci.99.10
ci.99.11
ci.99.12
ci.00.01
ci.00.02
ci.00.03
ci.00.04
ci.00.05
ci.00.06
ci.00.07
ci.00.08
ci.00.09
ci.00.10
ci.00.11
ci.00.12.
execute.
```

Appendix B

Microsoft Excel Visual Basic Macros

Used in Cleaning UCR Crime Data

Note: all of these macros were imbedded in separate spreadsheets in which column A contained the 2-letter state identifier.

First Macro:

```
Option Explicit
Option Base 1
Sub DoPreliminaries()

Dim iState As Integer, jLast As Integer
Dim k As Integer, kOut As Integer, x As Double, iBad As Integer
Dim iSht As Integer, iCol As Integer, iLast As Integer, jX As Integer
Dim startCol As Integer, iPage As Integer, iRow As Integer
Dim iMo As Integer, ColToFix As Integer

Dim stName As String, fromDir As String, toDir As String
Dim ss As String
Dim sORI As String, sCr As String
Dim sMo12 As String, stCr(8) As String

Dim wkMacro As Workbook, wkState As Workbook

Dim shtArr As Variant, shtArr1 As Variant, shtArr2 As Variant

    shtArr1 = Array("MU1", "RA1", "RO1", "AS1", "BU1", "LA1", "MV1", "CI1")
    shtArr2 = Array("MU2", "RA2", "RO2", "AS2", "BU2", "LA2", "MV2", "CI2")

    stCr(1) = "MU"
    stCr(2) = "RA"
    stCr(3) = "RO"
    stCr(4) = "AS"
    stCr(5) = "BU"
    stCr(6) = "LA"
    stCr(7) = "MV"

' Where do we find the state files, where do we store them,
' and do they have a "\" at their end? If not, add one.

    Set wkMacro = ActiveWorkbook
    fromDir = Range("E1")
    If Right(fromDir, 1) <> "\" Then fromDir = fromDir & "\"
    toDir = Range("E2")
    If Right(toDir, 1) <> "\" Then toDir = toDir & "\"

' Put times in column B, to see how long each state takes

    Range("B1") = Now

    For iState = 1 To 50

' Hide screen changes to make the computations go faster

        Application.ScreenUpdating = False

' Get the state file

        stName = Cells(iState + 1, 1)
        Workbooks.Open Filename:=fromDir & stName & "6A.xls"
```

```
        Set wkState = ActiveWorkbook

' Rename first sheet

        Sheets("extra").Select
        ActiveSheet.Name = "Extra"

' Clear extraneous information

        Range("AA:AA").Clear
        iLast = Range("K1").Value ' Number of rows

' Format the cells, resize the sheet, and freeze the panes

        For iSht = 2 To 18
            Sheets(iSht).Select
            Range("A1").Select
            Cells.NumberFormat = "General"
            ActiveWindow.FreezePanes = False
            ActiveWindow.LargeScroll ToRight:=-14
            Application.Goto Reference:=Cells(2, 2)
            If iSht = 2 Then Application.Goto Reference:=Cells(2, 3)
            With ActiveWindow
                .FreezePanes = True
                .Zoom = 75
            End With
        Next iSht

' If a crime count is less than -3, then record it in columns AA-AD
' and put the appropriate missing value in the cell.

        kOut = 2
        jX = 2
        For iSht = 3 To 16
            Sheets(iSht).Select
            For iRow = 1 To iLast
                For iCol = 2 To 145
                    If Cells(iRow, iCol) < -3 Then
                        iBad = Cells(iRow, iCol)
                        k = Int(iSht / 2 - 0.4) ' Crime type
                        With Cells(iRow, iCol)
                            .Value = -90 - k
                            .Interior.ColorIndex = 3
                        End With
                        With Sheets(18 - iSht Mod 2).Cells(iRow, iCol)
                            If .Value > -90 Then
                                .Value = -90 - k
                            Else
                                .Value = -98
                            End If
                            .Interior.ColorIndex = 3
                        End With
                        kOut = kOut + 1
                        With Sheets("Extra")
                            .Cells(kOut, 27) = iRow & ":" &
Sheets("First").Cells(iRow, 2)
                            .Cells(kOut, 28) = Sheets(iSht).Cells(1, iCol)
```

```
                        .Cells(kOut, 29) = iBad
                    End With
                End If
            Next iCol
        Next iRow
    Next iSht
    If kOut > 2 Then
        With Sheets(1)
            .Cells(1, 27) = "Negative Numbers"
            .Cells(2, 27) = "Row"
            .Cells(2, 28) = "Col"
            .Cells(2, 29) = "Value"
        End With
    End If

' Put -99 in where there is no "date updated" reported & color cell red

    For iRow = 2 To iLast
        For iCol = 171 To 194
            sMo12 = Worksheets("First").Cells(iRow, iCol)
            If sMo12 <> "111111111111" Then
                startCol = (iCol - 171) * 12 + 1
                If startCol >= 145 Then
                    startCol = startCol - 144
                    iPage = 2
                Else
                    iPage = 1
                End If
                For iMo = 1 To 12
                    If Mid(sMo12, iMo, 1) <> "1" Then
                        ColToFix = startCol + iMo
                        If iPage = 1 Then
                            Worksheets(shtArr1).Select
                        Else
                            Worksheets(shtArr2).Select
                        End If
                        If Worksheets("CI" & iPage).Cells(iRow, ColToFix) = 0
Then
                            Sheets("MU" & iPage).Cells(iRow, ColToFix).Select
                            With Selection
                                .Value = -99
                                .Interior.ColorIndex = 3 ' Red
                            End With
                        Else

' We get here if no date reported, but a number is found in
' the relevant cell on the CI page. It records the page,
' row, and column, all in column AE.

                            For k = 1 To 7
                                iBad = Sheets(stCr(k) & iPage).Cells(iRow,
ColToFix)
                                If iBad <> 0 Then Exit For
                            Next k
                            jX = jX + 1
                            Sheets("MU" & iPage).Cells(iRow, ColToFix).Select
                            With Selection
```

```
                                .Value = -90
                                .Interior.ColorIndex = 45 ' Orange
                            End With
                            Worksheets("Extra").Select
                            Cells(jX, 31) = iRow & ":" &
Sheets("First").Cells(iRow, 2)
                            Cells(jX, 32) = Sheets(stCr(k) & iPage).Cells(1,
ColToFix)
                            If k < 8 Then Cells(jX, 33) = iBad
                        End If
                    End If
                Next iMo
            End If
        Next iCol
    Next iRow

    If jX > 2 Then
        With Sheets(1)
            .Cells(1, 31) = "Misreported Crime"
            .Cells(2, 31) = "Row/ORI"
            .Cells(2, 32) = "Column"
            .Cells(2, 33) = "Value"
        End With
    End If

' Add in the missing Covered-by columns

    Sheets("First").Select
    Columns("BD:BD").Select
    Selection.Insert Shift:=xlToRight
    Range("BD1") = "COVER.80"
    Columns("BS:BS").Select
    Selection.Insert Shift:=xlToRight
    Range("BS1") = "COVER.95"
    Sheets("Extra").Select
    Range("A1").Select


' Save the file

    ActiveWorkbook.SaveAs toDir & stName & "7.xls"
    ActiveWindow.Close

' Turn screen back on

    Application.ScreenUpdating = True

    Range("B" & iState + 1) = Now
  Next iState
  ActiveWorkbook.Save
End Sub
```

Second Macro

```
Option Explicit
Option Base 1
Dim fromDir As String, toDir As String
Dim stName As String, iPage As Integer
Dim wkMacro As Workbook, wkState As Workbook
Dim shtAr1 As Variant, shtAr2 As Variant
Sub RunEachState()

Dim iSt As Integer

    shtAr1 = Array("MU1", "RA1", "RO1", "AS1", "BU1", "LA1", "MV1", "CI1")
    shtAr2 = Array("MU2", "RA2", "RO2", "AS2", "BU2", "LA2", "MV2", "CI2")

' Where do we find the state files, where do we store them,
' and do they have a "\" at their end? If not, add one.

    Set wkMacro = ActiveWorkbook
    fromDir = Range("E1")
    If Right(fromDir, 1) <> "\" Then fromDir = fromDir & "\"
    toDir = Range("E2")
    If Right(toDir, 1) <> "\" Then toDir = toDir & "\"

' Put times in column B, to see how long each state takes

    Range("B1") = Now

    For iSt = 1 To 50
        Call AnnualSemiQuarterly(fromDir, iSt)

' Save the file

        wkState.SaveAs toDir & stName & "8.xls"
        ActiveWindow.Close

' Turn screen back on

        Application.ScreenUpdating = True

' Enter time of completion

        Range("B" & iSt + 1) = Now

    Next iSt

' Save the macro file

    wkMacro.Save

End Sub
Sub AnnualSemiQuarterly(fromDir As String, iState As Integer)

Dim iYr1 As Integer, iYr2 As Integer, ss As String
Dim iMo As Integer, iYr As Integer
Dim sORI As String, jRow As Integer, sASQ As String
Dim kRow As Integer, iCol1 As Integer, iCol2 As Integer
```

```
Dim iRow As Integer, jLast As Integer, iPage As Byte
Dim Yr1 As Boolean, nCr(12) As Long, isNum(12) As Boolean

' Hide screen changes to make the computations go faster

    Application.ScreenUpdating = False

' Get the state file

    stName = Cells(iState + 1, 1)
    Workbooks.Open Filename:=fromDir & stName & "7.xls"
    Set wkState = ActiveWorkbook

' Find annual, semiannual, and quarterly indicators

    Worksheets("Extra").Select
    jLast = Range("L3").End(xlDown).Row
    For iRow = 3 To jLast

' Look at first character of column N, change to upper case

        Worksheets("Extra").Select
        sASQ = Range("N" & iRow)

' Sometimes the first character is a / followed by a or s or q
' If this is the case, remove the / and continue

        If Left(sASQ, 1) = "/" Then sASQ = Mid(sASQ, 2, 99)

        ss = UCase(Left(sASQ, 1))

        Select Case ss

' If A or S or Q

            Case "A", "S", "Q"

' First find the row corresponding to the ORI (column L)

                sORI = Sheets("Extra").Range("L" & iRow)
                Worksheets("First").Select
                kRow = Columns("B:B").Find(What:=sORI).Row

' What years does it apply to?

                Sheets("Extra").Select

                If Len(sASQ) = 1 Then ' Only for one year

' Get year in col. M to get columns

                        iYr1 = Year(Range("M" & iRow))
                        iYr2 = iYr1

                Else

' There may be more than 1 year specified
```

```
                    Call FindLimits(sASQ, iYr1, iYr2)
                End If
                Select Case ss

' For annual reports

                    Case "A"
                        Call Annual(kRow, iYr1, iYr2)

' For semiannual reports

                    Case "S"
                        Call SemiAnnual(kRow, iYr1, iYr2)
' For quarterly reports

                    Case "Q"
                        Call Quarterly(kRow, iYr1, iYr2)
                End Select
                With Sheets("Extra").Range("L" & iRow, "N" & iRow)
                    .Interior.ColorIndex = 8 ' Teal
                End With
        End Select
    Next iRow


    For iRow = 2 To Sheets(1).Cells(1, 11)
        Yr1 = True
        For iPage = 1 To 2
            For iYr = 1 To 11
                iYr1 = 1964 + iPage * 12 + iYr
                iCol1 = 12 * iYr - 10
                For iMo = 1 To 12
                    nCr(iMo) = Sheets("CI" & iPage).Cells(iRow, iCol1 + iMo - 1)
                    If nCr(iMo) < -99 Then Exit For 'Already aggregated
                    isNum(iMo) = (nCr(iMo) <> 0) And (nCr(iMo) > -4)
                    If isNum(iMo) And iMo Mod 3 <> 0 Then Exit For 'Crime in mos.
1, 2, 4, 5, 7, 8, 10, or 11
                Next iMo
                If iMo = 13 Then
                    If isNum(12) Then
                        If isNum(6) Then
                            If isNum(3) And isNum(9) And nCr(3) + nCr(9) > 5 Then
                                Call Quarterly(iRow, iYr1, iYr1)
                            Else
                                If nCr(6) > 4 Then Call SemiAnnual(iRow, iYr1,
iYr1)
                            End If
                        Else
                            If nCr(12) > 5 Then Call Annual(iRow, iYr1, iYr1)
                        End If
                    End If
                End If
                Yr1 = False
            Next iYr
        Next iPage
    Next iRow
```

```
End Sub
Sub Annual(iR As Integer, iY1 As Integer, iY2 As Integer)

Dim iY As Integer, iCol As Integer

    If iY2 < 1989 Then
        Worksheets(shtAr1).Select
        iPage = 1
        For iY = iY1 To iY2
            iCol = 12 * (iY - 1977) + 2
            Call ChangeCells(-112, iR, iCol, iCol + 10)
        Next iY
    ElseIf iY1 > 1988 Then
        Worksheets(shtAr2).Select
        iPage = 2
        For iY = iY1 To iY2
            iCol = 12 * (iY - 1989) + 2
            Call ChangeCells(-112, iR, iCol, iCol + 10)
        Next iY
    Else
        Worksheets(shtAr1).Select
        iPage = 1
        For iY = iY1 To 1988
            iCol = 12 * (iY - 1977) + 2
            Call ChangeCells(-112, iR, iCol, iCol + 10)
        Next iY
        Worksheets(shtAr2).Select
        iPage = 2
        For iY = 1989 To iY2
            iCol = 12 * (iY - 1989) + 2
            Call ChangeCells(-112, iR, iCol, iCol + 10)
        Next iY
    End If

End Sub
Sub SemiAnnual(iR As Integer, iY1 As Integer, iY2 As Integer)

Dim iY As Integer, iCol As Integer

    If iY2 < 1989 Then
        iPage = 1
        Worksheets(shtAr1).Select
        For iY = iY1 To iY2
            iCol = 12 * (iY - 1977) + 2
            Call ChangeCells(-106, iR, iCol, iCol + 4)
            Call ChangeCells(-112, iR, iCol + 6, iCol + 10)
        Next iY
    ElseIf iY1 > 1988 Then
        iPage = 2
        Worksheets(shtAr2).Select
        For iY = iY1 To iY2
            iCol = 12 * (iY - 1989) + 2
            Call ChangeCells(-106, iR, iCol, iCol + 4)
            Call ChangeCells(-112, iR, iCol + 6, iCol + 10)
        Next iY
    Else
```

```
        iPage = 1
        Worksheets(shtAr1).Select
        For iY = iY1 To 1988
            iCol = 12 * (iY - 1977) + 2
            Call ChangeCells(-106, iR, iCol, iCol + 4)
            Call ChangeCells(-112, iR, iCol + 6, iCol + 10)
        Next iY
        iPage = 2
        Worksheets(shtAr2).Select
        For iY = 1989 To iY2
            iCol = 12 * (iY - 1989) + 2
            Call ChangeCells(-106, iR, iCol, iCol + 4)
            Call ChangeCells(-112, iR, iCol + 6, iCol + 10)
        Next iY
    End If

End Sub
Sub Quarterly(iR As Integer, iY1 As Integer, iY2 As Integer)

Dim iY As Integer, iCol As Integer

    If iY2 < 1989 Then
        iPage = 1
        Worksheets(shtAr1).Select
        For iY = iY1 To iY2
            iCol = 12 * (iY - 1977) + 2
            Call ChangeCells(-103, iR, iCol, iCol + 1)
            Call ChangeCells(-106, iR, iCol + 3, iCol + 4)
            Call ChangeCells(-109, iR, iCol + 6, iCol + 7)
            Call ChangeCells(-112, iR, iCol + 9, iCol + 10)
        Next iY
    ElseIf iY1 > 1988 Then
        iPage = 2
        Worksheets(shtAr2).Select
        For iY = iY1 To iY2
            iCol = 12 * (iY - 1989) + 2
            Call ChangeCells(-103, iR, iCol, iCol + 1)
            Call ChangeCells(-106, iR, iCol + 3, iCol + 4)
            Call ChangeCells(-109, iR, iCol + 6, iCol + 7)
            Call ChangeCells(-112, iR, iCol + 9, iCol + 10)
        Next iY
    Else
        iPage = 1
        Worksheets(shtAr1).Select
        For iY = iY1 To 1988
            iCol = 12 * (iY - 1977) + 2
            Call ChangeCells(-103, iR, iCol, iCol + 1)
            Call ChangeCells(-106, iR, iCol + 3, iCol + 4)
            Call ChangeCells(-109, iR, iCol + 6, iCol + 7)
            Call ChangeCells(-112, iR, iCol + 9, iCol + 10)
        Next iY
        iPage = 2
        Worksheets(shtAr2).Select
        For iY = 1989 To iY2
            iCol = 12 * (iY - 1989) + 2
            Call ChangeCells(-103, iR, iCol, iCol + 1)
            Call ChangeCells(-106, iR, iCol + 3, iCol + 4)
```

```
            Call ChangeCells(-109, iR, iCol + 6, iCol + 7)
            Call ChangeCells(-112, iR, iCol + 9, iCol + 10)
        Next iY
    End If
End Sub
Sub ChangeCells(jVal As Integer, jR As Integer, _
    jC1 As Integer, jC2 As Integer)

' This subroutine changes the cell values to jVal and colors
' them green, for cells on row jR, from columns jC1 to jC2,
' on the pages ("1" or "2") specified by jPg
' It also colors the next cell (the one with the data) green

    Sheets("MU" & iPage).Range(Cells(jR, jC1), Cells(jR, jC2)).Select
    With Selection
        .Value = jVal
    End With
    Sheets("MU" & iPage).Range(Cells(jR, jC1), Cells(jR, jC2 + 1)).Select
    With Selection
        .Interior.ColorIndex = 4 ' Green
    End With

    End Sub

Sub FindLimits(sYrs As String, jY1 As Integer, jY2 As Integer)


' Possibly more than one year

    jY1 = Val(Mid(sYrs, 2, 99)) + 1900
    If jY1 = 1900 Then jY1 = 2000
    If Len(sYrs) > 6 Then ' More than one yr
        jY2 = Val(Right(sYrs, 2)) + 1900
        If jY2 = 1900 Then jY2 = 2000
    Else
        jY2 = jY1
    End If

End Sub
```

Third Macro

```
Option Explicit
Option Base 1
Dim fromDir As String, toDir As String
Dim stName As String, iCount As Integer
Dim wkMacro As Workbook, wkStFrom As Workbook, wkStTo As Workbook
Dim shtAr1 As Variant, shtAr2 As Variant
Sub RunEachState()

Dim iSt As Integer

    shtAr1 = Array("MU1", "RA1", "RO1", "AS1", "BU1", "LA1", "MV1", "CI1")
    shtAr2 = Array("MU2", "RA2", "RO2", "AS2", "BU2", "LA2", "MV2", "CI2")

' Where do we find the state files, where do we store them,
' and do they have a "\" at their end? If not, add one.

    Set wkMacro = ActiveWorkbook
    fromDir = Range("E1")
    If Right(fromDir, 1) <> "\" Then fromDir = fromDir & "\"
    toDir = Range("E2")
    If Right(toDir, 1) <> "\" Then toDir = toDir & "\"

' Put times in column B, to see how long each state takes

    Range("B1") = Now



    For iSt = 1 To 50
        Call CopyCoveredBys(fromDir, iSt)

' Save the file

'        ActiveWindow.Close

' Turn screen back on

'        Application.ScreenUpdating = True

' Enter time of completion

        Range("B" & iSt + 1) = Now

    Next iSt

' Save the macro file

    wkMacro.Save

End Sub
Sub CopyCoveredBys(fromDir As String, iState As Integer)

Dim iCr As Integer, iLast As Integer, rr As Range
Dim kRow As Integer, r As Range, x As Double
Dim iSht As Integer, iRow As Integer, iCol As Integer
```

```
' Hide screen changes to make the computations go faster

'    Application.ScreenUpdating = False


' Get the state file

    stName = Cells(iState + 1, 1)
    Workbooks.Open Filename:=fromDir & stName & "13B.xls"
    Set wkStFrom = ActiveWorkbook
    Workbooks.Open Filename:=toDir & stName & "8.xls"
    Set wkStTo = ActiveWorkbook

    wkStFrom.Sheets("First").Columns("BA:BX").Copy
    wkStTo.Sheets("First").Activate
    Columns("BA:BX").Select
    ActiveSheet.Paste
    ActiveWindow.FreezePanes = False
    Range("C2").Select
    Range("C2").Copy
    ActiveWindow.FreezePanes = True
    wkStFrom.Close savechanges:=False
    wkStTo.Close savechanges:=True

End Sub
```

Fourth Macro

```
Option Explicit
Option Base 1
Dim fromDir As String, toDir As String
Dim stName As String, iCount As Integer
Dim wkMacro As Workbook, wkState As Workbook
Dim shtAr1 As Variant, shtAr2 As Variant
Dim Crime(8) As String, iYr As Integer, iPg As Integer
Dim mRow As Integer
Sub RunEachState()

Dim iSt As Integer

    shtAr1 = Array("MU1", "RA1", "RO1", "AS1", "BU1", "LA1", "MV1", "CI1")
    shtAr2 = Array("MU2", "RA2", "RO2", "AS2", "BU2", "LA2", "MV2", "CI2")

    Crime(1) = "MU"
    Crime(2) = "RA"
    Crime(3) = "RO"
    Crime(4) = "AS"
    Crime(5) = "BU"
    Crime(6) = "LA"
    Crime(7) = "MV"
    Crime(8) = "CI"

' Where do we find the state files, where do we store them,
' and do they have a "\" at their end? If not, add one.

    Set wkMacro = ActiveWorkbook
    fromDir = Range("E1")
    If Right(fromDir, 1) <> "\" Then fromDir = fromDir & "\"
    toDir = Range("E2")
    If Right(toDir, 1) <> "\" Then toDir = toDir & "\"

' Put times in column B, to see how long each state takes

    Range("B1") = Now

    For iSt = 1 To 50
        stName = Cells(iSt + 1, 1)
        Workbooks.Open Filename:=fromDir & stName & "8.xls"
        Call CleanStateFiles(fromDir, iSt)

' Save the file

        wkState.SaveAs toDir & stName & "9.xls"
        ActiveWindow.Close

' Turn screen back on

        Application.ScreenUpdating = True

' Enter time of completion

        Range("B" & iSt + 1) = Now
```

```
    Next iSt

' Save the macro file

    wkMacro.Save

End Sub
Sub CleanStateFiles(fromDir As String, iState As Integer)

Dim iCr As Integer, rng As Range
Dim kRow As Integer, kLast As Integer
Dim iSht As Integer, i As Integer
Dim iRow As Integer, iLast As Integer
Dim iCol As Integer, iMo As Integer
Dim thisORI As String, thisDate As Date, thisCorr As String

' Hide screen changes to make the computations go faster

    Application.ScreenUpdating = False

' Get the state file

    Set wkState = ActiveWorkbook

' First, fix the annual data so only numbers > -4 are aggregated

    Sheets("Extra").Select
    Range("G2").Select
    ActiveCell.FormulaR1C1 = "=SUMIF(INDIRECT(RC[16]),"">-4"")/12"
    Selection.Copy
    Range(Selection, Selection.End(xlDown)).Select
    ActiveSheet.Paste

    iLast = Range("K1")

' Find last correction in column L

    kLast = Range("L3").End(xlDown).Row
    mRow = 2

    For kRow = 3 To kLast
        thisORI = Sheets("Extra").Range("L" & kRow)

' Find the row corresponding to the ORI

        iRow = Sheets("First").Range("B:B").Find(what:=thisORI).Row
        thisDate = Sheets("Extra").Range("M" & kRow)

' Find the year and page corresponding to the date

        iYr = Year(thisDate)
        iMo = Month(thisDate)
        Call GetCol(iPg, iYr, iMo, iCol)

' What's the correction message in column N?
```

```
        thisCorr = CStr(Sheets("Extra").Range("N" & kRow))

        Select Case UCase(Left(thisCorr, 1))
            Case "?", "!"
                Call Omit(kRow)
            Case "/"
                Call FixCells(kRow, iRow, iCol, iPg, Mid(thisCorr, 2, 99))
            Case "-"
                Select Case Val(thisCorr)
                    Case -1 ' Zero, presumed missing
                        If Sheets("CI" & iPg).Cells(iRow, iCol) > -99 Then
                            For i = 1 To 8
                                Call PaintCells(Crime(i), iRow, iCol, iCol, -98, 3)
' Red
                            Next i
                        End If
                    Case -12 To -2 ' Reported in another month
                        For i = 1 To 8
                            Call PaintCells(Crime(i), iRow, iCol, iCol, -100 +
thisCorr, 4) ' Green
                        Next i
                End Select
                Call Done(kRow)
            Case "A", "S", "Q"
            Case Else
                Call Omit(kRow)
        End Select
    Next kRow

' Reset the cursor to home on all sheets

    For iSht = 1 To 18
        Sheets(iSht).Activate
        Range("B2:B2").Select
    Next iSht
    If mRow > 2 Then
        With Sheets("Extra")
            .Cells(1, 31) = "Bad Values"
            .Cells(2, 31) = "Row"
            .Cells(2, 32) = "Column"
            .Cells(2, 33) = "Value"
        End With
    End If

End Sub
Sub Done(kRow As Integer)
    With Sheets("Extra").Range("L" & kRow, "N" & kRow)
        .Interior.ColorIndex = 4 ' Green
    End With
End Sub
Sub Omit(kRow As Integer)
    With Sheets("Extra").Range("L" & kRow, "N" & kRow)
        .Interior.ColorIndex = 3 ' Red
    End With
End Sub
Sub FixCells(kRow As Integer, iRow As Integer, iCol As Integer, _
    iPage As Integer, iCorr As String)
```

```
Dim s2 As String, iMo As Integer, i As Integer
Dim iBad As Double, iPg As Integer
Dim iCol1 As Integer, iCol2 As Integer
Dim iC As Integer, iM1 As Integer, iM2 As Integer


' We've already cut out the slash.
' Next, find the worksheet and column

    s2 = UCase(Left(iCorr, 2))
    Select Case s2
        Case "MU", "RA", "RO", "AS", "BU", "LA", "MV"

' Should never get here

            iBad = Val(Mid(iCorr, 3, 99))
            Sheets(s2 & iPage).Select

' Does the cell value match the value in column N?

            If iBad = Cells(iRow, iCol) Then ' Match
                For iC = 1 To 7
                    If s2 = Crime(iC) Then iC = 90 + iC
                Next iC
                Call PaintCells(s2, iRow, iCol, iCol, -iC, 45)  ' Orange
                Sheets("CI" & iPage).Select
                If Cells(iRow, iCol) > -80 Then
                    Call PaintCells("CI", iRow, iCol, iCol, -iC, 45)  ' Orange
                Else
                    Call PaintCells("CI", iRow, iCol, iCol, -98, 45)  ' Orange
                End If
                Call RecordBadValue(s2 & iPage, iBad, iRow, iCol)
                Call Done(kRow)
            Else
                Call Omit(kRow) 'No match
            End If
        Case "MO" ' Blank out contiguous months in a year
            If iPg = 1 Then
                Sheets(shtAr1).Select
            Else
                Sheets(shtAr2).Select
            End If
            Call Parse(Mid(iCorr, 3, 99), iM1, iM2)
            Call GetCol(iPg, iYr, iM1, iCol1)
            Call GetCol(iPg, iYr, iM2, iCol2)
            For i = 1 To 8
                Call PaintCells(Crime(i), iRow, iCol1, iCol2, -98, 3)
            Next i
            Call Done(kRow)
'        Case "A " ' These have been taken care of
        Case "X", "X " ' All zeros of this ORI are missing data
            Sheets(shtAr1).Select
            iPg = 1
            For iCol = 2 To 145
                If Sheets("CI1").Cells(iRow, iCol) = 0 Then
                    For i = 1 To 8
```

```
                        Call PaintCells(Crime(i), iRow, iCol, iCol, -98, 3)
                    Next i
                End If
            Next iCol
            Sheets(shtAr2).Select
            iPg = 2
            For iCol = 2 To 145
                If Sheets("CI2").Cells(iRow, iCol) = 0 Then
                    For i = 1 To 8
                        Call PaintCells(Crime(i), iRow, iCol, iCol, -98, 3)
                    Next i
                End If
            Next iCol
'        Case "O", "O " ' These will be taken care of
    End Select
End Sub
Sub Parse(stPars As String, iC1 As Integer, iC2 As Integer)
    iC1 = Val(stPars)
    iC2 = Val(Mid(stPars, InStr(1, stPars, "-") + 1, 99))
End Sub
Sub PaintCells(stCr As String, iR As Integer, iC1 As Integer, iC2 As Integer, _
    iVal As Integer, iColr As Integer)

    Sheets(stCr & iPg).Select
    Range(Cells(iR, iC1), Cells(iR, iC2)).Select
    With Selection
        .Interior.ColorIndex = iColr
        .Value = iVal
    End With

' Color the next cell if it aggregates other cells

    If iVal < -100 Then
        Sheets(stCr & iPg).Select
        Range(Cells(iR, iC2 + 1), Cells(iR, iC2 + 1)).Select
        With Selection
            .Interior.ColorIndex = iColr
        End With
    End If
End Sub
Sub GetCol(iPg As Integer, iYr As Integer, _
    iMo As Integer, iCol As Integer)

    If iYr > 1988 Then
        iPg = 2
        iCol = 12 * (iYr - 1989) + iMo + 1
    Else
        iPg = 1
        iCol = 12 * (iYr - 1977) + iMo + 1
    End If

End Sub
Sub RecordBadValue(stCr As String, badVal As Double, _
    xRow As Integer, xCol As Integer)

    mRow = mRow + 1
    With Sheets("Extra")
```

```
            .Cells(mRow, 31) = xRow & ":" & Sheets(stCr).Cells(xRow, 1)
            .Cells(mRow, 32) = Sheets(stCr).Cells(1, xCol)
            .Cells(mRow, 33) = badVal
        End With
    End Sub
```

Fifth Macro

```
Option Explicit
Option Base 1
Dim fromDir As String, toDir As String
Dim stName As String, iCount As Integer
Dim wkMacro As Workbook, wkState As Workbook
Dim shtAr1 As Variant, shtAr2 As Variant
Dim Crime(8) As String, iYr As Integer, iPg As Integer
Sub RunEachState()

Dim iSt As Integer

    shtAr1 = Array("MU1", "RA1", "RO1", "AS1", "BU1", "LA1", "MV1", "CI1")
    shtAr2 = Array("MU2", "RA2", "RO2", "AS2", "BU2", "LA2", "MV2", "CI2")

    Crime(1) = "MU"
    Crime(2) = "RA"
    Crime(3) = "RO"
    Crime(4) = "AS"
    Crime(5) = "BU"
    Crime(6) = "LA"
    Crime(7) = "MV"

' Where do we find the state files, where do we store them,
' and do they have a "\" at their end? If not, add one.

    Set wkMacro = ActiveWorkbook
    fromDir = Range("H1")
    If Right(fromDir, 1) <> "\" Then fromDir = fromDir & "\"
    toDir = Range("H2")
    If Right(toDir, 1) <> "\" Then toDir = toDir & "\"

' Put times in column B, to see how long each state takes

    Range("B1") = Now



    For iSt = 7 To 50

' Hide screen changes to make the computations go faster

    Application.ScreenUpdating = False

' Get the state file

        stName = Cells(iSt + 1, 1)
        Workbooks.Open Filename:=fromDir & stName & "9.xls"
        Call CheckORIExistence(fromDir, iSt)

' Save the file

        wkState.SaveAs toDir & stName & "10.xls"
        ActiveWindow.Close (False)

' Turn screen back on
```

```
        Application.ScreenUpdating = True

' Enter time of completion

        Range("B" & iSt + 1) = Now

    Next iSt

' Save the macro file

    wkMacro.Save

End Sub
Sub CheckORIExistence(fromDir As String, iState As Integer)

Dim iCt As Integer, jCt As Integer, iP As Integer
Dim iRow As Integer, iLast As Integer, iLine As Integer
Dim iCol As Integer, iC1 As Integer, iC2 As Integer
Dim iCol1 As Integer, iCol2 As Integer
Dim thisORI As String, thisDate As Date, thisCorr As String


    Set wkState = ActiveWorkbook
    iCt = 0
    iLine = 1

    iLast = Sheets("Extra").Cells(1, 11)
    For iRow = 2 To iLast
        For iCol = 5 To 28
            Sheets("First").Select
            If Len(Cells(iRow, iCol)) > 1 Then
                iC1 = iCol
                Exit For
            Else
                Call GetCol(iCol, iCol1, iCol2, iPg)
                Call PaintCells(iRow, iCol1, iCol2, -80, 33)
            End If
        Next iCol
        For iCol = 28 To 5 Step -1
            Sheets("First").Select
            If Len(Cells(iRow, iCol)) > 1 Then
                iC2 = iCol
                Exit For
            Else
                Call GetCol(iCol, iCol1, iCol2, iPg)
                Call PaintCells(iRow, iCol1, iCol2, -80, 33)
            End If
        Next iCol
        If iC1 < iC2 Then
            For iCol = iC1 To iC2
                Sheets("First").Select
                If Len(Cells(iRow, iCol)) < 2 Then
                    iCt = iCt + 1
                    With Sheets("Extra")
                        iLine = iLine + 1 ' Header line, ORI exists?
                        .Cells(iLine, 35) = iRow
```

```
                        .Cells(iLine, 36) = Cells(iRow, 2)
                        For iYr = 1977 To 2000
                            If Len(Cells(iRow, iYr - 1972)) > 0 Then
                            If .Cells(iLine, 37) = "" Then .Cells(iLine, 37) =
Cells(iRow, iYr - 1972)
                            .Cells(iLine, iYr - 1939) = "Exists"
                            End If
                        Next iYr
                        .Cells(3, 2) = iRow
                        iLine = iLine + 1 ' Crime data
                        .Cells(iLine, 37) = "Crime"
                        For iYr = 1977 To 2000
                            .Cells(iLine, iYr - 1939) = .Cells(iYr - 1975, 7) * 12
                        Next iYr
                        iLine = iLine + 1 ' Pop data
                        .Cells(iLine, 37) = "Population"
                        For iYr = 1977 To 2000
                            .Cells(iLine, iYr - 1939) = Cells(iRow, iYr - 1948)
                        Next iYr
                        iLine = iLine + 1 ' Covered-by data
                        For iYr = 1977 To 2000
                            .Cells(iLine, iYr - 1939) = Cells(iRow, iYr - 1924)
                        Next iYr
                    End With
' Check to see if a zero-pop agency
                    For iP = 29 To 52
                        If Cells(iRow, iP) > 0 Then
                            jCt = jCt + 1
                            Exit For
                        End If
                    Next iP
                    Exit For
                End If
            Next iCol
        End If
    Next iRow

    If iCt > 0 Then
        wkMacro.Sheets(1).Range("D" & iState + 1) = iCt
        With Sheets("Extra")
            For iYr = 1977 To 2000
                .Cells(1, iYr - 1939) = iYr
            Next iYr
            .Cells(1, 35) = "Line"
            .Cells(1, 36) = "ORI"
            .Cells(1, 37) = "Name"
        End With
    End If
    If jCt > 0 Then wkMacro.Sheets(1).Range("G" & iState + 1) = jCt

End Sub
Sub PaintCells(iR As Integer, iC1 As Integer, iC2 As Integer, _
    iVal As Integer, iClr As Integer)

    With Sheets("MU" & iPg)
        .Range(Cells(iR, iC1), Cells(iR, iC2)).Select
        With Selection
```

```
            .Interior.ColorIndex = iClr
            .Value = iVal
        End With
    End With
End Sub
Sub GetCol(iC As Integer, iC1 As Integer, iC2 As Integer, iP As Integer)

    If iC < 17 Then
        iC1 = 2 + (iC - 5) * 12
        iC2 = 13 + (iC - 5) * 12
        iP = 1
        Sheets(shtAr1).Select
    Else
        iC1 = 2 + (iC - 17) * 12
        iC2 = 13 + (iC - 17) * 12
        iP = 2
        Sheets(shtAr2).Select
    End If

End Sub
```

Sixth Macro

```
Option Explicit
Option Base 1
Dim fromDir As String, toDir As String
Dim stName As String, iCount As Integer
Dim wkMacro As Workbook, wkState As Workbook
Sub RunEachState()

Dim iSt As Integer


' Where do we find the state files, where do we store them,
' and do they have a "\" at their end? If not, add one.

    Set wkMacro = ActiveWorkbook
    fromDir = Range("E1")
    If Right(fromDir, 1) <> "\" Then fromDir = fromDir & "\"
    toDir = Range("E2")
    If Right(toDir, 1) <> "\" Then toDir = toDir & "\"

    For iSt = 1 To 50

' Hide screen changes to make the computations go faster

        Application.ScreenUpdating = False

' Get the state file

        stName = Cells(iSt + 1, 1)
        Workbooks.Open Filename:=fromDir & stName & "10.xls"
        Call DealWithOutliers(fromDir, iSt)

' Save the file

        wkState.SaveAs toDir & stName & "11.xls"
        ActiveWindow.Close False

' Turn screen back on

        Application.ScreenUpdating = True

' Enter time of completion


    Next iSt

' Save the macro file

    wkMacro.Save

End Sub
Sub DealWithOutliers(fromDir As String, iState As Integer)

Dim iPage As Integer, iCr As Integer
Dim iRow As Integer, jRow As Integer, iCt As Integer
Dim kCol As Integer, iLast As Integer
```

```
Dim iBad As Long, kBad As Long, iSame As Integer
Dim iMo As Integer, iYr As Integer

Dim sC(7) As String, thisORI As String, lastORI As String
Dim stFix As String, stCr As String
Dim stMo As String, stYr As String

Dim thisDate As Date, lastDate As Date

Dim crCell As Range, cXcell As Range

    sC(1) = "MU"
    sC(2) = "RA"
    sC(3) = "RO"
    sC(4) = "AS"
    sC(5) = "BU"
    sC(6) = "LA"
    sC(7) = "MV"
    iCt = 2
    iSame = 1

    Set wkState = ActiveWorkbook
    lastORI = ""
    lastDate = 0
    iLast = Sheets("Extra").Range("L3").End(xlDown).Row
    Sheets("Extra").Columns("AI:AL").Insert
    For jRow = 3 To iLast
        Sheets("Extra").Select
        If Cells(jRow, 12).Interior.ColorIndex <= 2 Then 'Skip colored cells
            thisORI = Cells(jRow, 12)
            thisDate = Cells(jRow, 13)
            If (thisORI <> lastORI Or thisDate <> lastDate) _
                And iSame < 7 And iSame > 1 Then
                iCt = iCt + 1
                With Sheets("Extra")
                    .Cells(iCt, 35) = iRow & ":" & thisORI
                    .Cells(iCt, 36) = iSame & " outlier crimes"
                End With
            End If
            iMo = Month(thisDate)
            If iMo < 10 Then stMo = "0" & iMo Else stMo = iMo
            iYr = Year(thisDate)
            stFix = Cells(jRow, 14)
            If UCase(Left(stFix, 2)) = "/O" Then 'Find outliers
                If thisORI = lastORI And thisDate = lastDate Then
                    iSame = iSame + 1
                Else
                    iSame = 1
                End If
                lastORI = thisORI
                lastDate = thisDate
                stCr = UCase(Mid(stFix, 4, 2))
                For iCr = 1 To 7
                    If stCr = sC(iCr) Then Exit For
                Next iCr
                If iYr < 1989 Then iPage = 1 Else iPage = 2
                stYr = Right(CStr(iYr), 2)
```

```
                On Error GoTo SkipIt
                iRow = Sheets("First").Columns("B:B").Find(What:=thisORI).Row
                kBad = Trim(Mid(stFix, 6, 99))
                kCol = (iYr - 1977 - (iPage - 1) * 12) * 12 + iMo + 1
                On Error GoTo SkipIt
                Set crCell = Sheets(stCr & iPage).Cells(iRow, kCol)
                iBad = crCell
                If iBad = kBad Then
                    ' Print it somewhere
                    With Sheets("Extra")
                        iCt = iCt + 1
                        .Cells(iCt, 35) = iRow & ":" & thisORI
                        .Cells(iCt, 36) = stCr & "." & stMo & "." & stYr
                        .Cells(iCt, 37) = iBad
                    End With
                    ' Paint the cells
                    Range("L" & jRow, "N" & jRow).Interior.ColorIndex = 24
                    ' Change the outlier cell to -9X
                    crCell = -90 - iCr
                    crCell.Interior.ColorIndex = 3
                    ' Change the CI cell to -9X or -98
                    Set cXcell = Sheets("CI" & iPage).Cells(iRow, kCol)
                    cXcell.Interior.ColorIndex = 3
                    If cXcell > -90 Then cXcell = -90 - iCr Else cXcell = -98
                End If
            End If
        End If
SkipIt:
    Next jRow
    If iCt > 1 Then
        With Sheets("Extra")
            .Cells(1, 35) = "Outliers"
            .Cells(2, 35) = "Row"
            .Cells(2, 36) = "Column"
            .Cells(2, 37) = "Value"
        End With
    End If
End Sub
```

Seventh Macro

```
Option Explicit
Option Base 1
Dim fromDir As String, toDir As String
Dim stName As String, iCount As Integer
Dim wkMacro As Workbook, wkState As Workbook
Sub RunEachState()

Dim iSt As Integer


' Where do we find the state files, where do we store them,
' and do they have a "\" at their end? If not, add one.

    Set wkMacro = ActiveWorkbook
    fromDir = Range("E1")
    If Right(fromDir, 1) <> "\" Then fromDir = fromDir & "\"
    toDir = Range("E2")
    If Right(toDir, 1) <> "\" Then toDir = toDir & "\"

    For iSt = 1 To 50

' Hide screen changes to make the computations go faster

        Application.ScreenUpdating = False

' Get the state file

        stName = Cells(iSt + 1, 1)
        Workbooks.Open Filename:=fromDir & stName & "11.xls"
        Call CheckCoveredByStatus(fromDir, iSt)

' Save the file

        wkState.SaveAs toDir & stName & "12.xls"
        ActiveWindow.Close

' Turn screen back on

        Application.ScreenUpdating = True

' Enter time of completion


    Next iSt

' Save the macro file

    wkMacro.Save

End Sub
Sub CheckCoveredByStatus(fromDir As String, iState As Integer)

Dim iCt As Integer, j As Integer, k As Integer
Dim iRow As Integer, iLast As Integer
Dim iCol As Integer
```

```
Dim NCovers As Integer, thisORI As String
Dim cRow As Integer, stOri(10) As String

    Set wkState = ActiveWorkbook
    iCt = 0

    iLast = Sheets("Extra").Cells(1, 11)
    Sheets("First").Select

' First, make room for lists of covering ORIs

    Sheets("Extra").Columns("AL:AO").Insert

    For iRow = 2 To iLast

' Next, get a listing of all covering ORIs for each ORI

        NCovers = 0
        For j = 1 To 10
            stOri(j) = ""
        Next j
        For iCol = 53 To 76
            If Len(Sheets("First").Cells(iRow, iCol)) > 2 Then
                thisORI = Sheets("First").Cells(iRow, iCol)
                If NCovers = 0 Then
                    NCovers = 1
                    stOri(1) = thisORI
                Else
                    For j = 1 To NCovers
                        If thisORI = stOri(j) Then Exit For
                    Next j
                    If j = NCovers + 1 Then
                        stOri(j) = thisORI
                        NCovers = j
                    End If
                End If
            End If
        Next iCol


' Print the list of covering ORIs in columns AM-AP

        With Sheets("Extra")
            If NCovers > 0 Then
                If .Range("AM1") = "" Then
                    .Range("AM1") = "Row:ORI"
                    .Range("AN1") = "Covering ORIs for this row"
                End If
                iCt = iCt + 1
                .Range("AM" & iCt + 1) = iRow & ":" & Sheets("First").Range("B" &
iRow)
                For k = 1 To NCovers
                    .Cells(iCt + 1, 39 + k) = stOri(k)
                Next k
            End If
        End With
```

```
        Sheets("Extra").Select
        Columns("AA:AA").AutoFit
        Columns("AE:AE").AutoFit
        Columns("AI:AI").AutoFit
        Columns("AM:AM").AutoFit
    Next iRow
    wkMacro.Sheets(1).Range("B" & iState + 1) = iCt

End Sub
```

Eighth Macro

```
Option Explicit
Option Base 1
Dim fromDir As String, toDir As String
Dim stName As String, iCount As Integer
Dim wkMacro As Workbook, wkState As Workbook
Sub RunEachState()

Dim iSt As Integer


' Where do we find the state files, where do we store them,
' and do they have a "\" at their end? If not, add one.

    Set wkMacro = ActiveWorkbook
    fromDir = Range("E1")
    If Right(fromDir, 1) <> "\" Then fromDir = fromDir & "\"
    toDir = Range("E2")
    If Right(toDir, 1) <> "\" Then toDir = toDir & "\"

    For iSt = 1 To 50

' Hide screen changes to make the computations go faster

        Application.ScreenUpdating = False

' Get the state file

        stName = Cells(iSt + 1, 1)
        Workbooks.Open Filename:=fromDir & stName & "11.xls"
        Call CheckCoveredByStatus(fromDir, iSt)

' Save the file

        wkState.SaveAs toDir & stName & "12.xls"
        ActiveWindow.Close

' Turn screen back on

        Application.ScreenUpdating = True

' Enter time of completion


    Next iSt

' Save the macro file

    wkMacro.Save

End Sub
Sub CheckCoveredByStatus(fromDir As String, iState As Integer)

Dim iCt As Integer, j As Integer, k As Integer
Dim iRow As Integer, iLast As Integer
Dim iCol As Integer
```

```
Dim NCovers As Integer, thisORI As String
Dim cRow As Integer, stOri(10) As String

    Set wkState = ActiveWorkbook
    iCt = 0

    iLast = Sheets("Extra").Cells(1, 11)
    Sheets("First").Select

' First, make room for lists of covering ORIs

    Sheets("Extra").Columns("AL:AO").Insert

    For iRow = 2 To iLast

' Next, get a listing of all covering ORIs for each ORI

        NCovers = 0
        For j = 1 To 10
            stOri(j) = ""
        Next j
        For iCol = 53 To 76
            If Len(Sheets("First").Cells(iRow, iCol)) > 2 Then
                thisORI = Sheets("First").Cells(iRow, iCol)
                If NCovers = 0 Then
                    NCovers = 1
                    stOri(1) = thisORI
                Else
                    For j = 1 To NCovers
                        If thisORI = stOri(j) Then Exit For
                    Next j
                    If j = NCovers + 1 Then
                        stOri(j) = thisORI
                        NCovers = j
                    End If
                End If
            End If
        Next iCol


' Print the list of covering ORIs in columns AM-AP

        With Sheets("Extra")
            If NCovers > 0 Then
                If .Range("AM1") = "" Then
                    .Range("AM1") = "Row:ORI"
                    .Range("AN1") = "Covering ORIs for this row"
                End If
                iCt = iCt + 1
                .Range("AM" & iCt + 1) = iRow & ":" & Sheets("First").Range("B" &
iRow)
                For k = 1 To NCovers
                    .Cells(iCt + 1, 39 + k) = stOri(k)
                Next k
            End If
        End With
```

```
'
'    With Sheets("First")
'        thisORI = .Range("BE" & iRow)
'        If Len(thisORI) > 2 Then
'            If .Range("BC" & iRow) = thisORI Then
'                .Range("BD" & iRow) = thisORI
'            Else
'                .Range("BD" & iRow).Interior.ColorIndex = 3
'            End If
'        End If
'        thisORI = .Range("BT" & iRow)
'        If Len(thisORI) > 2 Then
'            If .Range("BR" & iRow) = thisORI Then
'                .Range("BS" & iRow) = thisORI
'            Else
'                .Range("BS" & iRow).Interior.ColorIndex = 3
'            End If
'        End If
'    End With

        Sheets("Extra").Select
        Columns("AA:AA").AutoFit
        Columns("AE:AE").AutoFit
        Columns("AI:AI").AutoFit
        Columns("AM:AM").AutoFit
    Next iRow
    wkMacro.Sheets(1).Range("B" & iState + 1) = iCt

End Sub
```

Ninth Macro

```
Sub FixOutliersAndAddCountyData()

Dim stCr(7) As String, sTest As String
Dim stRow As String, stNot(7) As String
Dim i As Integer, j As Integer, k As Integer, bCr(7) As Byte
Dim iSt As Integer, lRow As Integer
Dim iPg As Integer, iCol As Integer
Dim iYr As Integer, iMo As Integer
Dim nOut As Integer, iRow As Integer, kRow As Integer
Dim shtAr1 As Variant, shtAr2 As Variant
Dim wkMacro As Workbook, wkState As Workbook, wkPop As Workbook
Dim rgOut As Range

    stCr(1) = "MU"
    stCr(2) = "RA"
    stCr(3) = "RO"
    stCr(4) = "AS"
    stCr(5) = "BU"
    stCr(6) = "LA"
    stCr(7) = "MV"

    shtAr1 = Array("MU1", "RA1", "RO1", "AS1", "BU1", "LA1", "MV1", "CI1")
    shtAr2 = Array("MU2", "RA2", "RO2", "AS2", "BU2", "LA2", "MV2", "CI2")

' Where do we find the state files, where do we store them,
' and do they have a "\" at their end? If not, add one.

    Set wkMacro = ActiveWorkbook
    fromDir = Range("D1")
    If Right(fromDir, 1) <> "\" Then fromDir = fromDir & "\"
    toDir = Range("D2")
    If Right(toDir, 1) <> "\" Then toDir = toDir & "\"
    Workbooks.Open Filename:=Range("D3")
    wkMacro.Activate

    For iSt = 1 To 50

        stname = Cells(iSt + 1, 1)
        Workbooks.Open Filename:=fromDir & stname & "12.xls"
        Set wkState = ActiveWorkbook
        wkState.Sheets("Extra").Select
        If Range("AJ3") <> "" Then

'How many outliers are there?

            lRow = Range("AJ2").End(xlDown).Row
            For iRow = lRow To 1 Step -1
                Sheets("Extra").Select
                nOut = Val(Left(Range("AJ" & iRow), 1))
                If nOut > 0 Then
                    Set rgOut = Range("AI" & iRow, "AK" & iRow)

'Make room for new outliers

                    If nOut < 6 Then
```

```
                    For j = 1 To 6 - nOut
                        rgOut.Insert (xlShiftDown)
                    Next j
                End If

'Which crime types are already accounted for? Denote by bCr = 1

                For j = 1 To 7
                    bCr(j) = 0
                Next j
                For j = iRow - nOut To iRow - 1
                    sTest = UCase(Left(Range("AJ" & j), 2))
                    For k = 1 To 7
                        If sTest = stCr(k) Then
                            bCr(k) = 1
                            Exit For
                        End If
                    Next k
                Next j

'Put the missing crimes in better order

                j = 1
                For k = 1 To 7
                    stNot(j) = ""
                    If bCr(k) = 0 Then
                        stNot(j) = stCr(k)
                        j = j + 1
                    End If
                Next k

'Now to fill in the missing cells and put missing values in cells

                iPg = 1
                iYr = Right(Range("AJ" & iRow - 1), 2)
                If iYr = 0 Then iYr = 100
                iMo = Mid(Range("AJ" & iRow - 1), 4, 2)
                iCol = (iYr - 77) * 12 + iMo + 1
                If iYr > 88 Then
                    iPg = 2
                    iCol = iCol - 144
                End If
                stRow = Range("AI" & iRow - 1)
                kRow = Left(stRow, InStr(1, stRow, ":") - 1)
                For j = 1 To 7 - nOut
                    Range("AI" & iRow + j - 1) = Range("AI" & iRow - 1)
                    Range("AJ" & iRow + j - 1) = stNot(j) & Right(Range("AJ" &
iRow - 1), 6)
                    Range("AK" & iRow + j - 1) = Sheets(stNot(j) &
iPg).Cells(kRow, iCol)
                Next j
                If iPg = 1 Then Sheets(shtAr1).Select Else
Sheets(shtAr2).Select
                With Sheets("MU" & iPg)
                    .Cells(kRow, iCol).Select
                    With Selection
                        .Interior.ColorIndex = 3
```

```
                        .Value = -98
                    End With
                End With
            End If
        Next iRow
    End If

'Now to deal with county designations

    Sheets.Add
    ActiveSheet.Name = "County data"
    Windows("Cty Pops w ORI.xls").Activate
    Sheets(stname).Activate
    Range(Cells(1, 1), Cells.SpecialCells(xlCellTypeLastCell)).Copy
    wkState.Activate
    Sheets("County data").Select
    Range("A1").PasteSpecial xlPasteValues ', Transpose:=True
    Range("A1").Select
    ActiveWindow.Zoom = 75

    Sheets("First").Select
    Columns("A:A").Clear
    Range("A1") = "COUNTY"
    If iSt > 1 Then
        For iRow = 2 To Sheets("Extra").Cells(1, 11)
            For iCol = 148 To 125 Step -1
                k = Val(Cells(iRow, iCol))
                If 0 < k < 275 Then
                    Cells(iRow, 1) = k
                    Exit For
                End If
            Next iCol
        Next iRow
    End If

    wkState.SaveAs toDir & stname & "13.xls"
    wkState.Close False
    wkMacro.Activate
  Next iSt
End Sub
Sub FindZeros()

Dim nZ As Integer
Dim wkMacro As Workbook, wkState As Workbook

  Set wkMacro = ActiveWorkbook
  toDir = Range("D2")
  If Right(toDir, 1) <> "\" Then toDir = toDir & "\"

  For iSt = 1 To 50

    stname = Cells(iSt + 1, 1)
    Workbooks.Open Filename:=toDir & stname & "13.xls"
    Set wkState = ActiveWorkbook
    Sheets("First").Select
    nZ = Application.WorksheetFunction.CountIf(Columns("A:A"), "=0")
    wkMacro.Sheets(1).Cells(iSt + 1, 2) = nZ
```

```
        wkState.Close False
    Next iSt

End Sub
```

Tenth Macro

```
Option Explicit
Option Base 1
Dim fromDir As String, toDir As String
Dim stName As String, iCount As Integer, nCover As Integer
Dim wkMacro As Workbook, wkState As Workbook
Public crChart As Chart, cbChart As Chart
Sub RunEachState()

Dim iSt As Integer, iAns As Integer


' Where do we find the state files, where do we store them,
' and do they have a "\" at their end? If not, add one.

    Set wkMacro = ActiveWorkbook
    fromDir = Range("E1")
    If Right(fromDir, 1) <> "\" Then fromDir = fromDir & "\"
    toDir = Range("E2")
    If Right(toDir, 1) <> "\" Then toDir = toDir & "\"
    iSt = ActiveCell.Row - 1
reStart:
    stName = UCase(InputBox("2-Letter State Designation ", _
        "Get State", Cells(iSt + 1, 1)))
    Cells(iSt + 2, 1).Select

' Get the state file

    On Error GoTo reTry
    iSt = Range("A:A").Find(stName).Row
    nCover = Cells(iSt, 2)
    Workbooks.Open Filename:=fromDir & stName & "13.xls"
    Set wkState = ActiveWorkbook
    Sheets("Extra").Activate
    Call PlotCoveredBys(fromDir, iSt)

    wkState.SaveAs (toDir & stName & "14.xls")
    wkState.Close
    wkMacro.Sheets(1).Range("A" & iSt).Interior.ColorIndex = 4



reTry:
    iAns = MsgBox("Another state?", vbYesNo)
    If iAns = vbYes Then GoTo reStart

End Sub
Sub PlotCoveredBys(fromDir As String, iState As Integer)

Dim iCt As Integer, j As Integer, k As Integer
Dim iRow As Integer, jRow As Integer, kRow As Integer
Dim iLast As Integer, iAns As Byte, iCB As Integer
Dim iPg As Integer, iC1 As Integer, iC2 As Integer

Dim thisORI As String, cvrORI(3) As String
Dim cRow As Integer, stOri(10) As String, stAns As String
```

```
Dim crName As String, cbName As String, topLine As String
Dim crRng As String, cbRng As String, sCB(24) As String
Dim stAll As String, stSome As String

Dim shtAr1 As Variant, shtAr2 As Variant

    shtAr1 = Array("MU1", "RA1", "RO1", "AS1", "BU1", "LA1", "MV1", "CI1")
    shtAr2 = Array("MU2", "RA2", "RO2", "AS2", "BU2", "LA2", "MV2", "CI2")

    iCt = 0

    iLast = Sheets("Extra").Cells(1, 11)
    jRow = 1

NextOne:

'First, what ORI is covered (listed in col. 39)?

    Sheets("Extra").Select
    jRow = jRow + 1
    If jRow > nCover + 1 Then GoTo theEnd
    thisORI = Cells(jRow, 39)
    j = InStr(1, thisORI, ":")
    iRow = Left(thisORI, j - 1)
    thisORI = Right(thisORI, 7)

'Which and how many coverers?

    For iCB = 1 To 3
        cvrORI(iCB) = Sheets("Extra").Cells(jRow, 39 + iCB)
        If cvrORI(iCB) = "" Then Exit For
    Next iCB
    iCB = iCB - 1

'Prepare to plot ORI & its coverer(s).

    With Sheets("Extra")
        .Range("C2:C289").Copy ' Dates
        .Range("Y2:Y289").PasteSpecial xlPasteValues
        Columns("Y").NumberFormat = "mmm-yy"

        Columns("Z:AC").Clear
        Columns("AE:AH").Clear

        .Range("Z1") = thisORI
        Sheets("CI1").Range("B" & iRow & ":" & "EO" & iRow).Copy
        .Range("Z2:Z145").PasteSpecial xlPasteValues, Transpose:=True
        Sheets("CI2").Range("B" & iRow & ":" & "EO" & iRow).Copy
        .Range("Z146:Z289").PasteSpecial xlPasteValues, Transpose:=True

        For k = 1 To 3
            Cells(1, 26 + k) = ""
        Next k
        For k = 1 To iCB
            kRow = Sheets("First").Range("B:B").Find(cvrORI(k)).Row
               If kRow > iLast Or kRow < 2 Then GoTo NextOne
```

```
            .Cells(1, 26 + k) = .Cells(jRow, 39 + k)
            Sheets("CI1").Range("B" & kRow & ":" & "EO" & kRow).Copy
            .Range("A" & Chr(64 + k) & "2").PasteSpecial xlPasteValues,
Transpose:=True
            Sheets("CI2").Range("B" & kRow & ":" & "EO" & kRow).Copy
            .Range("A" & Chr(64 + k) & "146").PasteSpecial xlPasteValues,
Transpose:=True
        Next k
        .Range("A1").Select
    End With

'Plot monthly crime data (ORI axis on LEFT).

    Charts.Add
    crRng = "Y1:A" & Chr(64 + iCB) & "289"
    Set crChart = ActiveChart
    With ActiveChart
        .ChartType = xlXYScatterLinesNoMarkers
        .SetSourceData Source:=Sheets("Extra").Range(crRng), PlotBy _
            :=xlColumns
        .Location Where:=xlLocationAsObject, Name:="Extra"
    End With
    With ActiveChart.Axes(xlCategory)
        .MinimumScale = 28126
        .MaximumScale = 36891
        .MinorUnit = 365.25
        .MajorUnit = 1461
        .MajorTickMark = xlCross
        .MinorTickMark = xlOutside
        .TickLabelPosition = xlNextToAxis
    End With
    ActiveChart.Axes(xlValue).MinimumScale = 0


    For k = 1 To iCB
        ActiveChart.SeriesCollection(k + 1).AxisGroup = 2
    Next k
    ActiveChart.Axes(xlValue, xlSecondary).MinimumScale = 0
    ActiveChart.HasLegend = True
    ActiveChart.Legend.Position = xlBottom
    crName = ActiveChart.Parent.Name

    ActiveSheet.ChartObjects(crName).Activate
    ActiveChart.ChartArea.Select
    ActiveSheet.Shapes(crName).IncrementLeft -250
    ActiveSheet.Shapes(crName).IncrementTop -150
    ActiveWindow.Visible = False


    For j = 1 To 24
        If Len(Sheets("First").Cells(iRow, j + 4)) > 0 Then
            Sheets("Extra").Cells(j + 1, 31) = 1
        Else
            Sheets("Extra").Cells(j + 1, 31) = 0
        End If
        For k = 1 To iCB
            If Sheets("First").Cells(iRow, j + 52) = cvrORI(k) Then
```

```
                Sheets("Extra").Cells(j + 1, 31 + k) = 1
            Else
                Sheets("Extra").Cells(j + 1, 31 + k) = 0
            End If
        Next k
    Next j

    Range("AE1") = thisORI
    Range("AF1:AH1").Clear
    For k = 1 To iCB
        Cells(1, 31 + k) = "CB " & cvrORI(k)
    Next k

    For j = 1 To 24
        If Cells(j + 1, 31) = 1 And (Cells(j + 1, 32) + _
            Cells(j + 1, 33) + Cells(j + 1, 34)) = 1 Then
            sCB(j) = "C"
        Else
            sCB(j) = "X"
        End If
        If sCB(j) = "X" Then
            Cells(j + 1, 30) = Right(CStr(j + 76), 2)
        Else
            Cells(j + 1, 30) = sCB(j)
        End If
    Next j
    Columns("AD").NumberFormat = "General"

    Charts.Add
    Set cbChart = ActiveChart
    cbRng = "AD1:A" & Chr(69 + iCB) & "25"
    ActiveChart.ChartType = xlColumnClustered
    ActiveChart.SetSourceData Source:=Sheets("Extra") _
        .Range(cbRng), PlotBy:=xlColumns
    ActiveChart.Location Where:=xlLocationAsObject, Name:="Extra"
    cbName = ActiveChart.Parent.Name
    ActiveChart.PlotArea.Width = 442
    ActiveChart.PlotArea.Left = 1
    ActiveSheet.Shapes(cbName).IncrementLeft -1500
    ActiveSheet.Shapes(cbName).IncrementTop 130
    ActiveChart.HasLegend = True
    ActiveChart.Legend.Position = xlBottom

    wkState.Activate
    Range("A22").Select

    topLine = ": No. " & jRow - 1 & " of " & nCover
    iAns = ShowFix(iRow, topLine)

    If iAns = vbNo Then
        stAll = ""
        For j = 1 To 24
            stAll = stAll & sCB(j)
        Next j
reDo:
        stAll = InputBox("Add/Subtract Covered Years", , stAll, 300, 2000)
        stAll = UCase(stAll)
```

```
        If Len(stAll) <> 24 Then
            MsgBox ("Incorrect length -- should be 24")
            GoTo reDo
        End If
        For j = 1 To 24
            If Mid(stAll, j, 1) <> "C" And Mid(stAll, j, 1) <> "X" _
                And Mid(stAll, j, 1) <> "M" Then Exit For
        Next j
        If j < 25 Then
            MsgBox ("Only X (not covered) or C (covered) or M (covered by month)
are OK")
            GoTo reDo
        End If
        For j = 1 To 24
                sCB(j) = UCase(Mid(stAll, j, 1))
            If sCB(j) = "X" Then
                Cells(j + 1, 30) = Right(CStr(j + 76), 2)
            Else
                Cells(j + 1, 30) = sCB(j)
            End If
        Next j
    End If
    ActiveSheet.ChartObjects(cbName).Delete
    ActiveSheet.ChartObjects(crName).Delete

'Here's where I have to replace the appropriate cells with -85
'and color them blue (20)

    For j = 1 To 24
        If sCB(j) <> "X" Then
            If j < 13 Then iPg = 1 Else iPg = 2
            iC1 = 12 * j - 144 * iPg + 134
            iC2 = iC1 + 11
            If iPg = 1 Then Sheets(shtAr1).Select Else Sheets(shtAr2).Select
            If sCB(j) = "C" Then
                With Sheets("MU" & iPg)
                    .Range(Cells(iRow, iC1), Cells(iRow, iC2)).Select
                    With Selection
                        .Interior.ColorIndex = 20
                        .Value = -85
                    End With
                End With
            ElseIf sCB(j) = "M" Then
                stSome = Sheets("First").Cells(iRow, 172 + j)
                For k = 1 To 12
                    If Mid(stSome, k, 1) = 0 Then
                        With Sheets("MU" & iPg)
                            .Range(Cells(iRow, iC1 - 1 + k), Cells(iRow, iC1 - 1 +
k)).Select
                            With Selection
                                .Interior.ColorIndex = 20
                                .Value = -85
                            End With
                        End With
                    End If
                Next k
            End If
```

```
        End If
    Next j



    GoTo NextOne

theEnd:
End Sub
Function ShowFix(iRow As Integer, sTitle As String) As String

Dim iAns As Byte, i As Byte, j As Byte, iPg As Byte
Dim iCol As Integer
Dim stIn As String, stOut As String
Dim stCov As String
Dim stLine(13) As String, allLines As String, yrLine As String

    yrLine = "
77|78|79|80|81|82|83|84|85|86|87|88|89|90|91|92|93|94|95|96|97|98|99|00" & Chr(13)
& Chr(10)
    stOut = wkMacro.Sheets(1).Cells(3, 6)
    stIn = wkMacro.Sheets(1).Cells(4, 6)
    For i = 1 To 12
        stLine(i) = i
        If i < 10 Then stLine(i) = "0" & i
    Next i
    For i = 1 To 24
        If i < 13 Then
            iCol = i * 12 - 11
            iPg = 1
        Else
            iCol = i * 12 - 155
            iPg = 2
        End If
        For j = 1 To 12
            If Sheets("CI" & iPg).Cells(iRow, iCol + j) > -3 Then
                stLine(j) = stLine(j) + stOut
            Else
                stLine(j) = stLine(j) + stIn
            End If
        Next j
    Next i
'    For i = 2 To 12
'        stLine(i) = stLine(i) '& Chr(13)
'    Next i
    For i = 2 To 6
        stLine(1) = stLine(1) & Chr(13) & stLine(i)
    Next i
    stLine(2) = ""
    For i = 7 To 12
        stLine(2) = stLine(2) & Chr(13) & stLine(i)
    Next i
    stLine(2) = stLine(2) & Chr(13) & yrLine

    ShowFix = MsgBox(stLine(1) & stLine(2), vbYesNo, _
        Sheets("First").Cells(iRow, 20) & sTitle & ": OK?")
```

```
End Function
```

Eleventh Macro

```
Option Explicit
Option Base 1
Dim wkMacro As Workbook, wkState As Workbook

Sub RunStates()

Dim fromDir As String, toDir As String, stName As String

Dim iSt As Integer

' Where do we find the state files, where do we store them,
' and do they have a "\" at their end? If not, add one.

    Set wkMacro = ActiveWorkbook
    fromDir = Range("D1")
    If Right(fromDir, 1) <> "\" Then fromDir = fromDir & "\"
    toDir = Range("D2")
    If Right(toDir, 1) <> "\" Then toDir = toDir & "\"
    wkMacro.Activate

    For iSt = 1 To 50
        If Cells(iSt + 1, 3) <> "Done" Then
            stName = Cells(iSt + 1, 1)
            Workbooks.Open Filename:=fromDir & stName & "14.xls"
            Set wkState = ActiveWorkbook

            Call CheckCounty(iSt)

            If iSt > 1 Then Call AggregateCrimeData(stName)

            wkState.SaveAs toDir & stName & "15.xls"

            wkState.Close False
            wkMacro.Activate
            Cells(iSt + 1, 3) = "Done"
        End If
    Next iSt
End Sub
Sub AggregateCrimeData(stNam As String)

Dim ctyRow As Integer, lastRow As Integer, ctyPlot As Integer
Dim iCty As Integer, ctyStart As Integer
Dim crCty(0 To 265, 1 To 288) As Long, iCr As Long, iDef As Integer
Dim iRow As Integer, iCol As Integer, iPg As Integer, thisCol As Integer
Dim iYr As Integer, iMo As Integer, jCr As Integer, k As Integer
Dim nCtys As Integer

'Dim tfChart As Boolean
Dim ctyChart As Chart
Dim x As String, crName As String, crType(10) As String

    crType(1) = "CI"
    crType(2) = "VI"
    crType(3) = "PR"
    crType(4) = "MU"
```

```
    crType(5) = "RA"
    crType(6) = "RO"
    crType(7) = "AS"
    crType(8) = "BU"
    crType(9) = "LA"
    crType(10) = "MV"

    wkState.Activate
    nCtys = Sheets("County data").Range("A3").End(xlDown).Row - 2
    Sheets("County data").Range("A2") = nCtys
    ctyStart = nCtys + 7

    lastRow = Sheets("Extra").Cells(1, 11)
'    lastCty = 0

    For jCr = 1 To 10
        If jCr > 1 Then ctyStart = ctyStart + 2 * nCtys + 5
        Sheets("County data").Cells(ctyStart - 1, 1) = crType(jCr)

'Aggregate ORI data to the county level into array crCty

        For iRow = 2 To lastRow
            If WorksheetFunction.IsNumber(Sheets("First").Cells(iRow, 1)) Then
' It's not a number if it registers as having 2 or more counties
                iCty = Sheets("First").Cells(iRow, 1)
'                If iCty > lastCty Then lastCty = iCty
                For iPg = 1 To 2
                    For iCol = 1 To 144
                        thisCol = iCol + 144 * (iPg - 1)
                        Select Case jCr
                            Case 1, 4 To 10
                                iCr = Sheets(crType(jCr) & iPg).Cells(iRow, iCol +
1)
                                If iCr > -4 Then crCty(iCty, thisCol) = _
                                    crCty(iCty, thisCol) + iCr
                            Case 2
                                For k = 4 To 7
                                        iCr = Sheets(crType(k) & iPg).Cells(iRow,
iCol + 1)
                                    If iCr > -4 Then crCty(iCty, thisCol) = _
                                        crCty(iCty, thisCol) + iCr
                                    Next k
                            Case 3
                                For k = 8 To 10
                                        iCr = Sheets(crType(k) & iPg).Cells(iRow,
iCol + 1)
                                    If iCr > -4 Then crCty(iCty, thisCol) = _
                                        crCty(iCty, thisCol) + iCr
                                    Next k
                        End Select
                    Next iCol
                Next iPg
            Else ' Check on which county for this year
                iPg = 1
                For iYr = 1 To 24
                    If iYr > 12 Then iPg = 2
                    iCty = Val(Sheets("First").Cells(iRow, 124 + iYr))
```

```
                    iCol = 12 * iYr - 144 * iPg + 132
                    For iMo = 1 To 12
                        iCol = iCol + 1
                        thisCol = iCol + 144 * (iPg - 1)
                        Select Case jCr
                            Case 1, 4 To 10
                                iCr = Sheets(crType(jCr) & iPg).Cells(iRow, iCol +
1)
                                If iCr > -4 Then crCty(iCty, thisCol) = _
                                    crCty(iCty, thisCol) + iCr
                            Case 2
                                For k = 4 To 7
                                    iCr = Sheets(crType(k) & iPg).Cells(iRow, iCol
+ 1)
                                    If iCr > -4 Then crCty(iCty, thisCol) = _
                                        crCty(iCty, thisCol) + iCr
                                Next k
                            Case 3
                                For k = 8 To 10
                                    iCr = Sheets(crType(k) & iPg).Cells(iRow, iCol
+ 1)
                                    If iCr > -4 Then crCty(iCty, thisCol) = _
                                        crCty(iCty, thisCol) + iCr
                                Next k
                        End Select
                    Next iMo
                Next iYr
            End If
        Next iRow

' Label counties

        ctyRow = ctyStart
        With Sheets("County data")
            For iCty = 0 To nCtys
                If iCty = 0 Then
                    .Cells(ctyRow, 1) = "No county"
                Else
                    .Cells(ctyRow, 1) = Left(.Cells(iCty + 2, 5), Len(.Cells(iCty +
2, 5)) - 7)  'Get rid of "county" or "parish"
                End If
                ctyRow = ctyRow + 1
                .Cells(ctyRow, 1) = "County " & iCty
                ctyRow = ctyRow + 1
            Next iCty

    ' Enter the data

            ctyRow = ctyStart
            For iCty = 0 To nCtys
                For iCol = 1 To 144
                    .Cells(ctyRow, iCol + 1) = crCty(iCty, iCol)
                    crCty(iCty, iCol) = 0
                Next iCol
                ctyRow = ctyRow + 1
                For iCol = 1 To 144
                    .Cells(ctyRow, iCol + 1) = crCty(iCty, iCol + 144)
```

```
                    crCty(iCty, iCol + 144) = 0
                Next iCol
                ctyRow = ctyRow + 1
            Next iCty
        End With
    Next jCr
'    tfChart = False
    iDef = 0
    Cells(1, 1).Select

End Sub
Sub CheckCounty(stNum As Integer)

' Some ORIs are listed in different counties. This macro records them in column
' 1 of "First," and colors them green. It may occur because a multi-county agency
' increases in size more in the secondary county. Or a new county is created (AZ).

Dim x As Integer, lastRow As Integer, i As Integer
Dim iRow As Integer, iCol As Integer
Dim nCty As Integer, iCty(24) As Integer, nMany As Integer

Dim stCty As String

Dim tfMany As Boolean, tfCty As Boolean

    wkState.Activate
    tfCty = False
    nMany = 0
    lastRow = Sheets("Extra").Cells(1, 11)
    Sheets("First").Select
    With Sheets("First")
        For iRow = 2 To lastRow
            stCty = .Cells(iRow, 1)
            tfMany = False
            For i = 1 To 24
                iCty(i) = 0
            Next i
            nCty = 0
            For iCol = 1 To 24
                    x = Val(.Cells(iRow, iCol + 124))
                    If x > 0 And x < 275 Then
                        If .Cells(iRow, 1) = 0 Then .Cells(iRow, 1) = x
                        If x <> .Cells(iRow, 1) Then
                            If nCty = 0 Then
                                stCty = stCty & ", " & x
                                tfMany = True
                                tfCty = True
                                nMany = nMany + 1
                                nCty = 1
                                iCty(1) = x
                            Else
                                For i = 1 To nCty
                                    If x = iCty(i) Then Exit For
                                Next i
                                If i > nCty Then
                                    stCty = stCty & ", " & x
                                    nCty = nCty + 1
```

```
                                iCty(nCty) = x
                            End If
                        End If
                    End If
                End If
            Next iCol
            If tfMany Then
                With .Cells(iRow, 1)
                    .Value = stCty
                    .Interior.ColorIndex = 4
                End With
            End If
        Next iRow
    End With

    If tfCty Then
        With wkMacro.Sheets(1)
            .Cells(stNum + 1, 1).Interior.ColorIndex = 4
            .Cells(stNum + 1, 2) = nMany
        End With
    End If

End Sub
Sub Fix300s()

' Some agencies have "300" for county number. this changes that to "0".

Dim fromDir As String, toDir As String, stName As String

Dim iSt As Integer, lRow As Integer, i As Integer, iCt As Integer

' Where do we find the state files, where do we store them,
' and do they have a "\" at their end? If not, add one.

    Set wkMacro = ActiveWorkbook
    fromDir = Range("D1")
    If Right(fromDir, 1) <> "\" Then fromDir = fromDir & "\"
    toDir = Range("D2")
    If Right(toDir, 1) <> "\" Then toDir = toDir & "\"
    wkMacro.Activate

    For iSt = 1 To 50
        If Cells(iSt + 1, 3) <> "Done" Then
            stName = Cells(iSt + 1, 1)
            Workbooks.Open Filename:=fromDir & stName & "14.xls"
            Set wkState = ActiveWorkbook

            iCt = 0
            Sheets("First").Select
            lRow = Range("A:A").End(xlDown).Row
            For i = 2 To lRow
                If Cells(i, 1) > 260 Then
                    Cells(i, 1) = 0
                    iCt = iCt + 1
                End If
            Next i
'            MsgBox (iCt)
```

```
            wkState.Close True
            wkMacro.Activate
            Cells(iSt + 1, 3) = "Done"
        End If
    Next iSt

End Sub
Sub GetORINames()

' This macro puts the last-used name of the ORI in column 4 of "First."

Dim fromDir As String, toDir As String, stName As String

Dim iSt As Integer, iRow As Integer, iCol As Integer, lastRow As Integer

' Where do we find the state files, where do we store them,
' and do they have a "\" at their end? If not, add one.

    Set wkMacro = ActiveWorkbook
    fromDir = Range("D1")
    If Right(fromDir, 1) <> "\" Then fromDir = fromDir & "\"
    toDir = Range("D2")
    If Right(toDir, 1) <> "\" Then toDir = toDir & "\"
    wkMacro.Activate

    For iSt = 1 To 50
        stName = Cells(iSt + 1, 1)
        Workbooks.Open Filename:=toDir & stName & "15.xls"
        Set wkState = ActiveWorkbook

        Sheets("First").Select
        Cells(1, 4) = "ORI NAME"
        lastRow = Sheets("Extra").Cells(1, 11)
        For iRow = 2 To lastRow
            For iCol = 28 To 5 Step -1
                If Len(Cells(iRow, iCol)) > 1 Then
                    Cells(iRow, 4) = Cells(iRow, iCol)
                    Exit For
                End If
            Next iCol
        Next iRow
        Range("D:D").Font.Bold = True

        wkState.Close True
    Next iSt
    wkMacro.Save
End Sub
```