Title

Teaching Introductory Geographical Data Analysis with GIS: A Laboratory Guide for an Integrated SpaceStat/Idrisi Environment (93-5)

Author

Dodson, Rusty (editor)

Publication Date

1993-05-01

NCGIA

National Center for Geographic Information and Analysis

Teaching Introductory Geographical Data Analysis with GIS:

A Laboratory Guide for an Integrated SpaceStat/ldrisi Environment

Edited by

Rusty Dodson
University of California, Santa Barbara

Technical Report 93-5

May 1993

Simonett Center for Spatial Analysis
University of California
35 10 Phelps Hall
Santa Barbara, CA 93106-4060
Office (805) 893-8224
Fax (805) 893-8617
ncgia@ncgia.ucsb.edu

State University of New York
301 Wilkeson Quad, Box 610023
Buffalo NY 14261-0001
Office (716) 645-2545
Fax (716) 645-5957
ncgia@ubvms.cc.buffalo.edu

University of Maine
348 Boardman Hall
Orono ME 04469-5711
Office (207) 581-2149
Fax (207) 581-2206
ncgia@spatial.maine.edu

TABLE OF CONTENTS

LAB EXERCISES

APPENDICES

PREFACE

This Technical Report contains a small number of laboratory exercises that were developed by Rusty Dodson in conjunction with an introductory course on geographical data analysis, offered to upper-level undergraduates in the Geography Department at the University of California, Santa Barbara. The course is intended to stress the spatial aspects of data used by geographers in empirical work. As part of an overall re-organization of the statistics curriculum in the department, I completely redesigned this course around an integrated framework of spatial (data) analysis and geographical information systems, in response to some of the research ideas generated as part of NCGIA Initiative 14 on Spatial Analysis and GIS. Given the introductory nature of the course, very little is assumed in terms of statistical pre-requisites. The course is intended to be a "second course in statistics", but while traditionally the "space" is only brought in at more advanced levels, I have chosen to put it up front.

The emphasis in the course is on displaying spatial data, looking for patterns and spatial associations, and on learning to think in spatial terms. Concepts such as spatial autocorrelation, spatial heterogeneity, contiguity, etc. are stressed from the start, in contrast to the usual practice. The course material includes simple descriptive statistics (including spatial means, etc.), measures of spatial autocorrelation, the analysis of scatterplots, and bivariate linear regression, with spatial ANOVA and trend surface regression as special cases. A more rigorous analysis of regression diagnostics is covered separately. The course is designed for a 10 week quarter system, which somewhat limits the range of material that can be covered.

In the laboratory exercises, the display and data retrieval are illustrated by means of the Idrisi GIS, while the statistical analysis is illustrated with *SpaceStat*. Most exercises closely follow the approach taken in the *SpaceStat* Tutorial (Luc Anselin, 1992, *SpaceStat Tutorial, A Workbook for Using SpaceStat in the Analysis of Spatial Data*, NCGIA Technical Software Series S-92-1) and in my Technical Report on *Spatial Data Analysis with GIS: An Introduction to Application in the Social Sciences* (NCGIA Technical Report 92-10), to which the interested reader is referred for technical details and further illustrations.

The development of the laboratory materials was funded in part through an Instructional Improvement Grant from the UC Santa Barbara Academic Senate.

Luc Anselin
Associate Director, NCGIA

Teaching introductory geographical data analysis with GIS: A laboratory guide for an integrated SpaceStat/Idrisi environment.

INTRODUCTION

This lab guide first describes the hardware and software configuration of the computer laboratory and the procedure for compiling student datasets. The next section contains the seven student laboratory exercises used in the course. The appendices contain instructions for installing data from the included diskette and ancillary information such as program documentation and tips for scanning and printing maps.

Much of the information in this report is tailored to the particular hardware/network/software configuration of the UCSB Geography computer lab. Thus in order to use this information in your own lab, certain modifications may need to be carried out. These modifications are straightforward if you have a working knowledge of the DOS operating system.

Software requirements

This lab guide requires that you have installed versions of Idrisi 4.0 or higher and SpaceStat 1.0 or higher. SpaceStat requires an IBM-compatible PC with a math coprocessor and at least a 386 CPU. For Idrisi, an EGA or VGA color monitor and a Microsoft-compatible mouse are required. See the Idrisi and SpaceStat manuals for complete hardware/software requirements.

Contents of the included diskette

The included diskette contains all of the data needed to run the labs in this guide as well as the custom software programs which link Idrisi and SpaceStat. In addition, the diskette contains the full text of this lab guide to facilitate editing of the labs. Appendix 1 contains instructions for installing data and programs from the diskette. See the READ.ME file on the diskette for the latest information on the contents.

NOTE: The Columbus, Ohio dataset referenced in the lab exercises and included on the diskette is the same dataset which is provided with the *SpaceStat* package and referenced in the *SpaceStat* tutorial (Anselin, 1992). The version on the diskette includes subtle changes such as different variable names.

LABORATORY SOFTWARE AND HARDWARE

The laboratory section for the 172A course took place in Geography's PS/2 laboratory, Ellison Hall 2612, and makes use of ten IBM PC computers networked to an RS/6000 fileserver. Printing support is provided by an IBM postscript printer. The lab also has a scanner, two digitizers, and a CD-ROM reader. These facilities (scanner, digitizers, CD-ROM) were not used in the 172A course for Fall, 1992, but they might be useful in compiling datasets in the future.

For numerical analysis in the lab, students use SpaceStat, a spatial statistical package written by Luc Anselin here at UCSB Geography. SpaceStat is currently the only comprehensive software package available which addresses a wide range of spatial statistical methods. However, SpaceStat provides no facility for mapping or statistical graphics. For graphical support, students use Idrisi, a grid-based geographic information system (GIS) developed by

Ron Eastman at Clark University. Idrisi is a low-cost, easy-to-use software package which is widely used in teaching GIS.

In order to use SpaceStat as an "analytical engine" and Idrisi as a "graphical engine" in the laboratory, it was necessary to write a set of linkage programs which allow data to be passed between the two software packages. The linkage software is written in both C and DOS batch languages, and is fully documented in Appendix 2. Source code for the linkage programs is provided on the included diskette.

Management of data

Each of the lab exercises in this document is written for a specific dataset. In some cases this dataset must be used; in some cases the dataset serves as an example and students are encouraged to do the lab on their own data. In order to maintain the integrity of the original datasets, a master copy is stored in read-only format on the fileserver, which appears to students as the **G: drive**, and students use a copy of the dataset in their work directory **D:\TEMP**. When students start work in the lab, the D:\TEMP directory is completely erased, and a clean copy of the dataset is installed from the read-only directory G:\U\NCGIA\172SETUP. This way, there are no problems with corrupted datasets, and data volumes on the local hard drives are minimized since all student work takes place in a single directory which is constantly purged.

Students copy datasets using a command called **172LOAD**, which takes one of the following arguments: **lab1**, **lab2**, **lab3**, **lab4**, **lab5**, **lab6**, **lab7**, **mydata**. The "mydata" argument assumes students are using their own data, and copies only the setup files necessary to run SpaceStat and the linkage programs. The other arguments copy the setup files as well as a dataset for that particular lab exercise.

When students are finished working in the lab, they need to save important data files to their diskettes, since the D:\TEMP work directory will be erased by the next student. Since Idrisi image files require a lot of storage space, a command called MAPVAL was written for this course which allows Idrisi **values files** to be easily mapped with a single command. Values files are much smaller than image files and can be written directly by SpaceStat. The procedure for writing a .VAL with SpaceStat is contained in lab 2, under the section "Creating .VAL files for variables in a dataset". These files have a .VAL extension, and are the key to transferring information between Idrisi and SpaceStat. By using .VAL files to store data to be mapped, students need only store one Idrisi image on their diskette. This image should be called POLYID, and should contain an integer polygon identifier (poly-ID) which corresponds to each observation in the student's dataset. Thus, the dataset should have a variable called **ID** which contains poly-ID values matching those of the POLYID Idrisi image. The observations in the dataset should be sorted on the **ID** variable.

Given a .VAL file, the MAPVAL command can be used to display a map of the variable contained in the .VAL file. The MAPVAL command calls two executable C programs which do some preprocessing[1], and it uses Idrisi commands to assign the variable values in the .VAL file to the polygons in the POLYID image, making a new image called TEMP1. The TEMP1 image is constantly overwritten, but can be saved by using the Idrisi MAINT command to rename it.

---

[1] The C programs create a .DVC values documentation file (VAL_PREP.EXE), and write a legend to the .DOC image documentation file (AUTOLEG.EXE).

Thus the MAPVAL command allows any desired data in SpaceStat to be moved to Idrisi. In order to move data in the opposite direction (from an Idrisi image to SpaceStat), the Idrisi command EXTRACT is first used to extract a .VAL file from the image. Then a C program (VAL2SPC.EXE) is used to add the header information required by SpaceStat. The modified .VAL file can then be easily read in to SpaceStat using the "Data/Input/ASCII to dataset" option (**D-1-1**), and merged with an existing dataset using the "Data/Merge/Select/Add variables" option (**D-3-1**). The SpaceStat dataset should be sorted on the poly-ID variable.

Conventions

The SpaceStat/Idrisi linked environment assumes the following conventions:

1. Students work in a directory called D:\TEMP, which contains all Idrisi and SpaceStat data, as well as certain required setup files such as IDRISI.ENV and IDRISI.PRN. A full set of the required setup files is contained in the directory \172SETUP\MYDATA on the included diskette.

2. The integer polygon identifiers (poly-IDs) are stored in SpaceStat as a variable called **ID**. The SpaceStat dataset should be sorted on this variable in increasing numeric order. The poly-ID must be integer, but does not have to be numbered continuously and does not have to start with 1. The upper limit on a poly-ID is 32767, which is Idrisi's limit for an integer.

3. Students have an Idrisi image called **POLYID**, which contains poly-IDs which match exactly the values of the **ID** variable in the SpaceStat dataset.

4. All necessary programs (Idrisi executables and the programs from the 172BIN directory on the included diskette) are located somewhere on the DOS search path. See the next section for more information on the search path.

Computer configuration

To configure computers in the PS/2 lab for the 172A course, students type **space**, and if necessary, re-boot the machine. The "space" configuration installs a minimum number of devices[1] in order to maximize the available memory, and sets up the DOS search path to include the directory containing Idrisi (e.g. G:\U\LAB\IDRISI), and a directory called 172BIN (e.g. G:\U\NCGIA\172BIN) which contains executable linkage programs. The full search path used during Fall 1992 is:

```
C:\BAT;G:\U\NCGIA\172BIN;G:\U\LAB\IDRISI;C:\DOS
```

Note that SpaceStat is not on the above path. The 172BIN directory contains a program called **SPACE.BAT** which needs to contain the correct drive/path on which you have installed SpaceStat. This procedure is described in the installation section of this report.

STUDENT DATASETS

The spatial statistical techniques covered in this course focus on areal data observations, referenced by polygons, however many of the techniques are appropriate for other types of data as well. Students are encouraged to use their own data for their labs and final projects.

---

[1] A mouse is required for running Idrisi. Additional devices provide network and printer access.

The G172A course at UCSB used the 1:2,000,000 scale ARCUSA CDROM[1] as a source for creating small student datasets. The ARCUSA CDROM contains hundreds of demographic, political, environmental, and agricultural variables for the 3,111 counties of the contiguous United States, and was quite successful at giving students a wide variety of choices in terms of the geographic area and the type of data to be studied. The next section describes the procedure for creating student datasets from the CDROM.

If you don't have access to the ARCUSA CDROM or another such data repository, your students will need to obtain some Idrisi-compatible polygon data and corresponding statistical data. Statistical data should be abundant in any library. A relatively simple way to obtain Idrisi polygon data is to scan hard-copy maps, or boundary lines traced from an atlas. Assuming you have access to a scanner which produces .TIF files, Appendix 5 describes a procedure for converting a "raw" scanned map into a clean set of Idrisi polygons.

Extracting data from the ARCUSA CD-ROM

First, students are given a listing of the available variables in the ARCUSA database and a map of the 3,111 US counties (Appendix 7). Then they are instructed to pick a set of five to ten related variables of interest from the variable list and to pick a contiguous study area of 50 to 100 counties from the map. They choose counties by either shading them in on the map or by providing a written description, e.g. "fifty counties on the Missouri/Kansas border centered around Kansas City". A list of variables and a description of counties is given to the TA, who then extracts the requested information from the CDROM and converts it into a dataset compatible with SpaceStat and Idrisi.

Given a county/variable data request, a couple of Arc/Info routines must be run in order to provide two things: a raster file of the county polygons, and a flat ASCII file of the variables, referenced by a polygon identifier (poly-ID). Following the procedure outlined in Appendix 6 produces both of these things. Once this is done, a complete student dataset is composed of three files: an ASCII file of variables (DATA.ASC), and an Idrisi image consisting of the files POLYID.IMG and POLYID.DOC. These three files make up the point of departure for Lab 2, Introduction to SpaceStat.

The complete procedure for extracting student datasets from the ARCUSA CD-ROM is outlined in Appendix 6. Source code for the extraction macros is on the included diskette.

STUDENT EXERCISES

The next section contains the text of seven student exercises (labs). Each lab is designed to build on the material covered in the previous lab.

NOTE: These labs refer to specific disk drives and directories and a specific software/hardware/network situation which may not match your lab situation.

REFERENCES

Anselin, L. 1992. "SpaceStat: a program for the analysis of spatial data." *NCGIA Software 92-S-1*, Santa Barbara: National Center for Geographic Information and Analysis.

---

[1] The ARCUSA database was provided by Environmental Systems Research Institute (ESRI), 380 New York Street, Redlands, CA 92373-9870.

Eastman, J. R. 1992. *IDRISI Version 4.0*, Worcester, Massachusetts: Clark University Graduate School of Geography.

Environmental Systems Research Institute, 1991. *Arc/Info 6.0 Command References*, Redlands: ESRI.

DOS NOTES[1]

This handout describes a few simple DOS concepts and commands.

DISK DRIVES

The machines you are using each utilize several *disk drives*, which are referenced by a letter followed by a colon, viz: A:, C:, D:, G:, H:. Different drives are accessed by typing the appropriate drive name. For example, typing:

```
D:
```
[2]

would move you to the "D" drive. Each disk drive contains its own directory and file structure. On the machines in this lab, the A: drive is the *floppy drive*. It cannot be accessed unless there is a readable floppy diskette in the disk drive.

Drives "C" and "D" are the *hard disk drives*, which exist inside the computer and are always available.

Drives "G" and "H" are the *server drives*. These drives do not exist on the PS/2 machines--they exist on a remote machine (known as the file *server*) and are accessed through a *network*. Due to the unpredictable nature of the network in this lab, you may periodically find that these drives are not available. If this is the case, try *rebooting* the computer by pressing the "Ctrl", "Alt", and "Del" keys simultaneously. If two or three reboots don't fix the problem, and you need to access the server drives, you need to move to another machine.

DIRECTORIES AND FILES

Directories: In DOS, *files* are used to store data and program modules like the ones you will be using to run IDRISI and SpaceStat. Related files are grouped together into *directories*. Directories are analogous to file folders, and files are analogous to the individual documents stored within a given folder. In addition to files, a directory may also contain other directories. These are referred to as *subdirectories*. The existence of subdirectories means that files can be organized into a hierarchical structure. The highest-level directory in this hierarchy is referred to as the *root directory*.

Directory navigation

The CD command is used to navigate the hierarchy. This command changes the *current directory*. The current directory is the directory whose files and subdirectories are directly accessible to you.

To change the current directory, the CD command is entered along with the desired directory name. For example, if the current directory contains a subdirectory called LABS, then

```
CD  LABS
```

---

[1] Originally written by Howard Veregin, UC Santa Barbara.

[2] DOS is *case insensitive,* which means that it doesn't distinguish between upper and lower-case letters. You can type commands in either manner.

causes LABS to become the current directory. If the current directory does not contain this subdirectory, then the entire "pathname" can be specified. For example,

```
CD  \COURSE\LABS
```

changes the current directory to LABS regardless of the current location in the hierarchy. The backslash "\" appearing as the first character of the pathname tells CD to start at the root directory. For this example to work properly, LABS must be a subdirectory of COURSE, and COURSE must be a subdirectory of the root directory.

The backslash "\" character is always used with DOS commands to separate directory names on a given pathname. Do not confuse it with the conventional slash character "/".[1]

Special pathnames also exist. For example,

```
CD  ..
```

moves the current directory up one level in the hierarchy, while

```
CD  \
```

moves the current directory to the root directory.

Listing files in a directory

The DIR command is used to list the contents of a specified directory. For example,

```
DIR  LABS
```

lists the contents of LABS (assuming that LABS is a subdirectory of the current directory). The entire pathname may also be specified. For example,

```
DIR  \COURSE\LABS
```

lists the contents of LABS regardless of the current directory. If DIR is entered without specifying a directory, the contents of the current directory are listed.

If the list of contents is too long to fit on the screen, the /P flag can be used.

```
DIR  LABS  /P
```

causes DIR to pause when the screen is full. The listing is resumed by depressing any key.

Creating/deleting directories

Directories are created using the MD (Make Directory) command and removed using the RD (Remove Directory) command. In order to remove a directory, it must be empty (i.e., contain no files or subdirectories). The MD and RD commands have the same format as DIR.

---

[1] For you Unix buffs, the DOS directory structure is essentially the same as that used by Unix, except that DOS uses a backslash "\" and Unix uses a forward slash "/" as the directory delimiter.

Working with Files

DOS has special requirements for naming files. The "filename" itself cannot be longer than eight characters, and certain characters (e.g., "?" and "*") are not permitted. Filenames can be followed by an "extension" up to three characters in length. However, extensions are not required by DOS. Some extensions (e.g., .EXE, .BAT and .COM) have special significance to DOS and are probably best avoided. The filename and extension are separated by a period (with no spaces in between).

Viewing the contents of a file

In DOS, commands that operate on files require you to specify the filename and extension (if any) in addition to the pathname. If the pathname is omitted the current directory is assumed. For example, the command

```
TYPE  LAB1.TXT
```

lists the contents of the file called LAB1.TXT in the current directory. The command

```
TYPE  \COURSE\LABS\LAB1.TXT
```

lists the contents of the file regardless of the current directory.

The TYPE command is useful for short files, but for files that overflow a single screen, MORE is preferable. MORE is not a command but a *filter*. To use it, you must redirect data into the filter using the redirection arrow (i.e., <). For example, to list the contents of LAB1.TXT, you would enter

```
MORE  <  LAB1.TXT
```

Hit any key to continue the listing. If you don't want to view the entire file, you can use CTRL+C to abort the listing (Hold down the "Ctrl" key while pressing the "c" key).

Another useful command is DEL. This command deletes the specified file, and has the same format as TYPE.

Copying files

The COPY command is used to make a duplicate copy of a specified file. For example,

```
COPY  LAB1.TXT  BADLAB1.TXT
```

creates a new file called BADLAB1.TXT that is a duplicate of the existing file LAB1.TXT.

Wildcards

When you refer to filenames in DOS, the characters "*" and "?" act as *wildcards* in the sense that they match more than one character. The star "*" character matches zero or more occurrences of any character (any letter or number). Thus, if you type:

```
DIR *.IMG
```

the resulting screen will list all files that end with the ".IMG" extension.

The "?" character matches a single occurrence of any character. Typing the command:

```
DEL LAB?.DOC
```

would delete the files "LAB1.DOC", LAB2.DOC", "LABX.DOC", etc., assuming those file existed in the first place. As a final example, typing:

```
DIR LAB???.A*
```

would match the following files:

```
LABABC.AB
LABXXX.A
LAB123.AAA
LABWEE.AW
LAB_1_.A_2
etc.
```

DISKETTES

As disk space is limited, it is good practice to clean up your account by deleting all unnecessary files, including those used in the previous assignment. If you want to keep copies of certain files, you can transfer them to a *floppy disk* (also known as a *diskette*) before deleting them from the hard disk drive.

Floppy disk drives are most often referred to as the A: or the B: drive. One common type of floppy drive accepts 3.5-inch double-sided high-density diskettes. These hold 1.44 megabytes of data. (A megabyte (Mb) equals about one million bytes.) To determine whether or not your diskette has enough space for the files you want to transfer, CD to the directory containing the files, and then type DIR. The third column in the resulting list of files indicates the size of each file in bytes. You may need to use more than one diskette if you are transferring several large files.

Diskettes must be *formatted* before data can be stored on them. Assuming your floppy drive is the A: drive, formatting is achieved by inserting the diskette into the drive and typing

```
FORMAT  A:
```

Note that this will destroy any data previously contained on the diskette, so only do it the first time you use the diskette. The system will inform you when formatting has been completed.

Use the COPY command to transfer a file to the diskette. Include the A drive as part of the second pathname. For example,

```
COPY  LAB1.TXT  A:\LAB1.TXT
```

will transfer a file called LAB1.TXT in the current directory. To retrieve your diskette, push the little blue button next to the disk drive.

LAB 0: Dos exercise[1]

INTRODUCTION

In this assignment you will be using a few simple DOS commands to navigate directories and examine files stored on the PS/2's. The commands you will be using are all described in the accompanying handout. You should complete the assignment even if you are already familiar with DOS.

List and view files

1. Boot up the system and, once the DOS prompt appears, change the current directory to G:\U\LAB\IDRISI. This is where IDRISI's program modules are stored. Use the DIR command to list the contents of the directory. List six different filename extensions that are used for the files in this directory.

2. Enter the command DIR *.PAL. What does this do? What is the purpose of the * character?

3. Use TYPE or MORE to view the contents of the file called READ.ME. What is in this file?

4. Copy READ.ME to a new file called GEOGRAPHY.TXT in the C:\TEMP directory (if C:\TEMP doesn't already exist, create it). Now use DIR to list the new file. What has happened to the filename? Delete the file.

---

[1] Originally written by Howard Veregin, UC Santa Barbara.

Directory navigation

5. Move to the C:\TEMP directory and create a subdirectory called JUNK. How is it represented by the DIR command?

6. Move to the JUNK subdirectory. How does DIR inform you that it is empty? What are the "." and ".." directories?

7. From the JUNK subdirectory, copy the READ.ME file in IDRISI to a file called DONTREAD.ME in JUNK. You will have to specify the pathname of READ.ME so that DOS can find it. What does DIR show now?

8. Move back to the IDRISI directory and try to remove JUNK. How does DOS respond?

9. Move to the JUNK subdirectory again and delete DONTREAD.ME. Repeat step 8. Were you successful this time?

LAB 1: Introduction to the Idrisi GIS[1]

INTRODUCTION

This exercise introduces concepts in Geographic Information Systems (GIS), by providing an opportunity for you to explore Africa through the Global Change Africa Database and Idrisi GIS software. After completing this lab, you should feel comfortable using Idrisi commands for displaying and manipulating spatial data. You will be using Idrisi throughout the quarter to support your geographic data analyses.

Setting up your data directory

All of your work in this course will take place in the D:\TEMP directory. This is a temporary directory, and will be erased frequently. Therefore, if you wish to save any files between lab sessions, you must copy them to your diskette using the DOS "COPY" command.

STEP 1: loading data

The following command will copy all the data you need for lab 1 into your D:\TEMP directory. At the dos prompt, type the following:

```
172load   lab1
```

> NOTE: The first time you use the 172LOAD command, you might have to include its complete path, e.g. "E:\LAB\STATPROG\172LOAD LAB1".

Note the space between "...172load" and "lab1". This command will warn you that it is about to delete all files in the D:\TEMP directory. If this is OK, type "y". You should then see the names of a number of files being copied to the D:\TEMP work directory. When the copying is completed, you will see a message which says to stay in this directory while working on the lab.

STEP 2: verifying your setup

Type the Idrisi command **LIST**. If everything was copied correctly, you should see a listing of available Idrisi images. If you don't see this, ask for help. You should now be ready to go through the lab exercise. In the following pages, you will be instructed to use various Idrisi commands, such as **COLOR**, **COLOR A**, **HISTO**, and **RECLASS**. For the purposes of this class, all commands will be run by simply typing the command name at the DOS prompt. (You might be accustomed to using Idrisi's menu interface. We will not be using the Idrisi menus in this class.)

---

[1] Originally created by Margaret Livingston and Rusty Dodson with assistance from Howard Veregin. NCGIA-Santa Barbara, February, 1991.

SECTION ONE

Viewing the Data

Listing the available images

To begin with, we will look at the themes, or *images*, in the Global Change Database. Use Idrisi's LIST module[1] to display a list of the themes in the database. Each theme is stored as a separate image file. LIST will display the name of each image[2] followed by a descriptive title. View the entire list to get a sense of what types of themes are included.

Displaying individual images

Use the **COLOR** module (type "color" at the DOS prompt) to view LANDUSE, an image of land use classes, or categories. Following Idrisi's prompts, enter **3** to choose the IBM color palette. Accept the default settings for all other questions by pressing **Enter**. Press **page down** and **page up** to scroll through all of the legend items. When you have finished viewing the image, press **Esc** to return to the DOS prompt.

Now view TMPJAN, mean January temperature, using **COLOR A**. (**COLOR A** automatically scales the range of temperature values into sixteen classes for display. The **COLOR** module (without the "A" option) cannot properly display an image that has more than sixteen values.) Choose the Idrisi color palette and accept the defaults for all other questions.

Adding vector files to image displays

The database also contains several *vector*[3] files which can be added to the images displayed on the screen. While the TMPJAN image is on the screen, press **v** to indicate that you wish to add a vector file. Following the prompts in the lower left screen corner, enter the vector filename **coasts** and a color code (a number from 0-15. Try 0 first.) such that the vector boundaries are visible. Press **v** again and display the vector file **country** in the same manner.

Viewing Surfaces

GIS allows us to look at and manipulate data in ways that paper maps do not. When looking at a paper map, you are limited to that view of the data. You cannot change how the data are represented without making a new map. In a GIS, however, data can easily be viewed in more than one way.

Use **COLOR A** to display ELEV (elevation and bathymetry). This view is similar to the way a paper map would display elevation.

---

[1] Here, "module" means the same thing as "command".

[2] Image names are limited to 8 characters. Whenever possible, the image names in the database attempt to describe the content of the image. This is important because you will be constantly creating new images when working with IDRISI.

[3] IDRISI *vector* files store boundary information as lists of x,y coordinate pairs. IDRISI image files use the *raster* data structure, which stores information as a regular grid of square or rectangular cells, each of which contains a numerical value.

Three-dimensional displays with ORTHO

Another way to visualize the same data is with a perspective view. Use **ORTHO** to view ELEV. Press **n** when Idrisi asks whether you want to drape another image, and accept all of Idrisi's default parameters by pressing **[Enter]** at the remaining prompts.

The three-dimensional view can be enhanced by assigning different colors to different groups or classes of elevations. This can be done by "draping" another image over the 3-D surface image. Idrisi requires that the draped image file have at most sixteen values. However, the ELEV image contains elevation values which range over several thousands of feet. Therefore, you must create a "drape-able" elevation image by grouping these elevation values into sixteen classes.

Creating an image suitable for draping

To create such an image file, use the **RECLASS** module. The image to be reclassified is ELEV. Call the output image file ELEV16. Choose the **equal intervals classification** option. Do not reset the minimum and maximum values. Then choose the **specifying the number of classes to be used** option, and enter **16** as the number of classes. The title of the new image should be something like "16 elevation classes". Press **[Enter]** in response to "value units".

Now create a three-dimensional display of ELEV using **ORTHO** as you did before, but this time tell Idrisi that you want to drape ELEV16 over the surface. Use the *Idrisi* color palette.

Statistical Analysis

GIS can also be used for statistical calculations which cannot be performed easily using paper maps. Use **COLOR** to display WIND, using the *Idrisi* palette. This image shows average windspeed in 13 categories from 0-1 to >12 meters per second.

Displaying an image histogram

Now use **HISTO** to create and display a histogram of WIND . Change the minimum value to **1** (which screens out the background areas that are coded with zeros), but leave the maximum value at **13**. Enter **1** for the class width. Select the *graphic* output mode. Each bar on the resulting histogram represents one wind speed category. The vertical axis, labeled "f", is the relative frequency of each category (i.e., the number of cells in that category which appear in the image).

* Are all windspeed categories equally frequent? Which category is the most frequent? What range of windspeeds does this category represent? Can you discern anything from the histogram about the locations at which the different windspeed categories occur?

Measuring areas

Another common GIS function is the measurement of areas on an image.

Use **COLOR** with the *IBM* palette to display VEG, vegetation types (again, **page down** and **page up** will scroll though all of the legend items).

* Try to determine the most common and least common vegetation type from the display.

Most common type:

Least common type:

Using the Idrisi **AREA** module, calculate areas for each vegetation type in VEG. Select **tabular** output in **cells** (cells refer to individual units of data on the image--the Idrisi images you are using each contain over 13,000 cells).

* Which vegetation type covers the most cells? Which vegetation type covers the least?

Most common type (most cells):

Least common type (least cells):

Investigating Relationships between Themes

Suppose that you wanted to examine the relationship between rainfall and elevation in Africa. One way to do this visually would be to display elevation as a three-dimensional surface and then to drape a rainfall image over it.

First use **RECLASS** to reclassify PRANN (mean annual precipitation) into an output file called PRECIP16. Follow the same procedure that you used earlier to create ELEV16.

Now use **ORTHO** to display ELEV as a 3-D surface. Drape PRECIP16 over the surface using the *Idrisi* color palette.

* Can you see any relationship between rainfall and elevation? Is rainfall higher in valleys or on mountains? Is it higher on the eastern or western slopes of mountains? What other themes in the database might also be related to elevation?

Cell Values

Idrisi allows us to find the value of a unit of data at a specific point. Display TMPANN, mean annual temperature, using **COLOR A**. Add the vector files COASTS and COUNTRY as you did before by pressing **v**.

Viewing an individual cell value with the screen cursor

Now press **c** to make a cursor appear on the screen. Use the mouse to move the cursor around on the image, and note that the column and row numbers for the cell at the center of the cursor are given in the lower left corner of the screen. Move the cursor to an area of interest, and click the **left** mouse button (this will freeze the cursor's position). The "z-value" (displayed at lower left next to the image coordinates) is the average annual temperature in units of 0.1 degrees Celsius. Thus, for example, a z-value of 154 = 15.4 degrees C. Click the **left** mouse button again to unfreeze the cursor, and view as many cell values as you wish. When finished, click the **right** mouse button to remove the cursor.

Zooming in on an image

Display any desired image using **COLOR** or **COLOR A**. Here's how to zoom in on an area of interest: press **w** (for "zoom Window") to bring up a screen cursor. Move the cursor to one corner of the desired zoom window and click the **left** button. Move the cursor to another window corner and click the **right** button. Note that you can display vector files on the zoomed image by pressing **v** as you did before.

To zoom out to the entire image, hit the **w** key again, and click the **right** button (without hitting the left button at all).

SECTION TWO (OPTIONAL)

If you wish to explore the capabilities of Idrisi in more detail, the following exercise will guide you through a simple analysis of two variables: wind speed and landuse.

Investigating Wind Speed and Soil Erosion

Suppose that you are studying the effects of wind on soil erosion. Areas which are on the margins of deserts and are prone to high wind speeds are susceptible to erosion. Thus, you want to identify these areas as regions which may be in danger of losing potentially valuable soil resources.

Use **DESCRIBE** on WIND to find the cell values for average windspeeds greater than 3 meters per second. These will be considered moderate to severe winds. Using **RECLASS** on WIND, create a new image called HIGHWIND. Choose "User-defined classification". Assign a new value of **1** to windspeeds greater than 3 meters/second (old values from 4 to 999) and a new value of **0** to all other windspeeds (old values from 0 to <4).

Next, use **DESCRIBE** on VEG2 to determine the value of the desert category and note the range of category numbers. Use **RECLASS** on VEG2 to create a new image called DESERT which shows only desert areas. To do this, assign the desert cells a value of **1**, and all other cells a value of **0**.

Now, HIGHWIND and DESERT will be combined into one image by overlaying the two images and adding their cell values. Think about what this process will do.

Use **OVERLAY** to add HIGHWIND and DESERT. Choose the **add** option and call the output image HIGHDES. Display HIGHDES with **COLOR**.

> Interpret the HIGHDES image. What do the cell values mean? (Remember that the input images contain only 0's and 1's.)

Use **OVERLAY** again but this time choose the **multiply** option to create HIGHDES2 from HIGHWIND and DESERT. Display HIGHDES2.

> How does this image compare with HIGHDES? What is the same and what is different? What caused this difference?

LAB 2: Introduction to SpaceStat[1]

This lab guides you through some introductory procedures with SpaceStat, the spatial statistics package you will be using throughout the quarter. For most of the spatial analytical techniques used in this class, SpaceStat will be used for the number-crunching statistical tasks, and Idrisi will be used for graphical display and other spatial data handling tasks.

SETTING UP YOUR DATA

1. Follow your computer lab's procedure for configuring the computer for SpaceStat and Idrisi.

2. You should now be in the D:\TEMP directory. Load data for this lab by typing:

   172load lab2

   This will copy a set of the data you need to the D:\TEMP work directory. Once this data is copied, you are ready to work with SpaceStat. In addition, you can type Idrisi commands at the DOS prompt. The lab data includes an Idrisi image called POLYID, which contains identifier numbers for polygons which represent neighborhoods of Columbus, Ohio. The example data set in SpaceStat corresponds to these Columbus polygons. You can display this image with Idrisi's **COLOR** command.

3. Type **space** to start up SpaceStat.

A QUICK LOOK AT SPACESTAT

The menu structure

SpaceStat is organized into four modules: DATA, TOOLS, EXPLORE, and REGRESS. Each module has two or three menus of functions below it. You navigate between modules by pressing the "hot keys" **D**, **T**, **E**, and **R**. These may be upper or lower case. Try pressing these keys, and note how the menu on the screen changes. In order to run one of the numbered functions on the screen menu, you can press the appropriate number key on the keyboard, or you can use the arrow keys to move the highlight bar to the appropriate number and then press **[Enter]**.

In these labs you will see individual SpaceStat functions referenced by a letter-number combination which refers to that function's location in the menu structure. For example, when the lab says "Use **D-1-1** to go to Data/Input/ASCII to dataset", this means you should press **D** to go to the DATA module, select menu option 1 (a menu of input functions) and select option 1 from the next menu, which is the ASCII-to-dataset function.

The option screen: [F1]

Pressing **F1** takes you to a screen of options which control various aspects of SpaceStat such as file output. These options are accessed just like SpaceStat menu items: you can type the number of the option, or you can use the highlight bar and press **[Enter]**. Some options

---

[1] By Rusty Dodson, NCGIA Santa Barbara, 9/15/92.

require that you type in something, and some simply toggle between YES and NO when you access them. When you have the options set the way you want them, press **Esc** to get back to the regular menu. NOTE: when you quit SpaceStat, all options return to their defaults. They are not saved.

Exiting to DOS: [F2]

In many instances you will want to run DOS commands (e.g. DIR, COPY) or Idrisi commands (e.g. COLOR, HISTO) while you are using SpaceStat. To do this without quitting SpaceStat, press **F2**. You will then be at the familiar DOS prompt, from which you can run DOS commands. When you want to return to SpaceStat, type **exit** at the DOS prompt.

NOTE: If you try to run a large program (for example, a word-processing program) while SpaceStat is running, you may get the error "out of memory" or something similar. If this happens, you must type **exit** to return to SpaceStat, and then quit SpaceStat by typing **Alt+Q**. This will remove SpaceStat from memory and allow you to run the other program.

IDRISI NOTE: Certain Idrisi 4.0 commands might not work properly when SpaceStat is in memory. If unexpected results occur, quit SpaceStat and then try the Idrisi command again.

Quitting SpaceStat: [Alt+Q]

To quit SpaceStat, type **Alt+Q** while any menu is displayed (but don't quit now).

CREATING A SPACESTAT DATASET FROM AN ASCII FILE

SpaceStat stores data in a proprietary binary format. In order to use your own data with SpaceStat, you must convert it to this format. In this section you are given a "raw" set of data in flat ASCII format. This is one of the most widely-known standard formats for the transfer of data. The ASCII dataset will be converted to the format required by SpaceStat.

The ASCII dataset file is called COLUMBUS.ASC. Press **F2** to go to DOS and look at the first few lines of the file by typing:

```
more < columbus.asc
```

The first part of this ASCII file contains information about the dataset, and the rest of the file contains the data values themselves. The first two numbers in the file refer to the number of observations (49) and the number of variables (7) in the dataset. The next part of the file contains the names for each of the variables, thus in this case there are seven names. The rest of the file contains the data values, presented in the same order as that of the variable names. Now that you have verified that the ASCII file is in the correct format for input to SpaceStat, the following procedure will convert the ASCII file into a SpaceStat dataset.

1. Type **exit** to return to SpaceStat and use **D-1-1** to go to Data/Input/ASCII to dataset.

2. Enter **columbus.asc** as the ASCII input file, and enter **columbus** as the dataset filename.

3. If all goes well, SpaceStat will report that a dataset called COLUMBUS was created with 49 observations on 7 variables. Press **[Enter]** to get back to the menu.

4. Press **F2** to escape to DOS. Type **dir columbus.*** and note that two new files have been created: COLUMBUS.DAT and COLUMBUS.DHT. These two files together make up a SpaceStat dataset. This fact is important to remember because when you copy datasets to your diskette, you must copy both the .DAT and the .DHT files.

5. Type **exit** to get back to SpaceStat.

CREATING .VAL FILES FOR VARIABLES IN A DATASET

In order to move information between Idrisi and SpaceStat, a simple file structure known as a *values file* will be used. Values files have the extension .VAL, and contain two columns of data: the first column is a polygon identification number (polygon-ID), and the second is a data value. In this course the polygon-ID will always be referenced as a variable called **ID**.

Given the information in a values file, a variable in SpaceStat can be passed to Idrisi for mapping, and a variable created by Idrisi can be passed to SpaceStat for statistical analysis. The following procedure shows you how to make .VAL files for three variables (crime, income, and housing value) in the COLUMBUS dataset you just created.

1. In order to make .VAL files from SpaceStat, the "Idrisi interface" option must be turned on. Press the **F1** key to go to the options screen, and change the Idrisi interface option (option number **5**) to "YES". Now press **Esc** to return to the regular SpaceStat screen.

2. Use **D-8-3** to go to the "list selected variables" function. Type the dataset filename **columbus**, and then type the variable names as shown below:

   NOTE: THE **ID** VARIABLE MUST BE ENTERED FIRST so that the .VAL files are sorted properly. Otherwise, your data will be garbled.

   Variable name: **id**
   Variable name: **crime**
   Variable name: **income**
   Variable name: **housing**
   Variable name: **[Enter]** (finished typing variable names)

3. SpaceStat now asks for the output format. Type "**0**" at the first prompt for the default format. Then press **[Enter]** a couple of times to clear the screen display.

4. Since you set the "Idrisi interface" option, SpaceStat used the first variable you entered (ID) as the polygon-ID for each of the values files. Three values files were therefore created, each with the ID variable in the first column, and CRIME, INCOME, or HOUSING in the second column. To verify that the files now exist, press the **F2** key to escape to DOS, and type:

```
dir   *.val
```

You should see that the files CRIME.VAL, INCOME.VAL, and HOUSING.VAL exist in your work directory. The next section explains how to map the above variables with Idrisi.

MAPPING .VAL FILES WITH IDRISI

1. Look at the polygon-IDs for Columbus by typing, at the DOS prompt:

col[1] **polyid**

Press **[Page Down]** a few times and you should see all 49 polygons, 16 at a time, get shaded based on their polygon-ID number. (Idrisi can only display 16 colors on the screen at one time.) Press **[Esc]** to clear the map.

3. Now you are going to attach a variable (stored in a .VAL file) to the polygons in the POLYID image. Type the following command:

   mapval

   You should now see the command *usage* for the MAPVAL command. MAPVAL (a mnemonic for "map a values file") expects an Idrisi image called POLYID to exist, and it requires that you give it the name of an existing .VAL file. Now, type the following:

   mapval crime

   After a few seconds, you should see a map of the CRIME variable for Columbus displayed with a blue and red color palette. While the map is displayed on the screen, you can perform several additional functions:

* Press "**v**", enter the name **colvec**, and the color code "**0**". This displays a vector file of polygon boundaries (COLVEC) with color zero (black).

* Press "**p**" and enter the palette name **grey**. Press "**p**" again and enter **blured** to go back to the original palette. Other valid palette names are **idrisi** and **ibm**.

* Press "**c**" and a cursor will appear on the screen. Using the mouse, click **left** to query the value of a particular polygon. Click **right** to remove the cursor.

4. Use MAPVAL to look at the other variables INCOME and HOUSING. How are the three variables distributed across space? Do you see any spatial patterns or possible relationships among the variables?

When you are finished, type **exit** (if necessary) to return to SpaceStat and then **Alt-Q** to quit.

---

[1] **COL** is a shortcut for Idrisi's **COLOR** command. **COLA** is another shortcut program, which runs Idrisi's **COLOR A** command.

LAB 3: Exploratory Data Analysis[1]

INTRODUCTION

This lab guides you through some basic exploratory data analysis (EDA) techniques on the Columbus, Ohio dataset you used last week. The lab focuses on a single variable, CRIME, and analyzes the variable by means of maps, histograms, and descriptive statistics. The purpose of the lab is to use these exploratory techniques to "get a feel" for the data--in terms of both the spatial distribution (by mapping the data) and the statistical distribution (by using histograms and descriptive stats).

Your homework assignment is to perform, *on your own data*, all of the procedures outlined in this lab. A complete description of the homework is provided at the end of this lab, along with tips on preparing your data.

HOMEWORKS ARE DUE ONE WEEK FROM TODAY

SETTING UP YOUR WORK DIRECTORY

1. Follow your computer lab's procedure for configuring the computer for SpaceStat and Idrisi.

2. You should now be in the D:\TEMP directory. Load data for this lab by typing:

   172load lab3

   This will copy a set of the data you need to the D:\TEMP work directory. Once this data is copied, you are ready to run SpaceStat, Idrisi commands, and various linkage programs.

PART ONE - SIMPLE STATISTICS AND GRAPHICS

Listing the variables in a dataset

1. Run SpaceStat by typing "**space**". When you see the startup screen, press any key to start the program.

2. Use **D-8-1** to go to the "summary information on data set" function. (The "D-8-1" means: type "**D**" to go to the data module; then type "**8**" to go to option number 8; then type "**1**" to go to option number 1.)

3. Now press "**Enter**", and SpaceStat will list all available datasets. There should be only one, called "columbus". Type "**columbus**" at the "Enter file name:" prompt, and you will see a summary of the variables in the Columbus dataset. This summary dataset operation is not really necessary as far as data analysis is concerned; it is included here because it is a useful function that you should be familiar with.

---

[1] By Rusty Dodson, NCGIA Santa Barbara, 10/19/92.

Make a .VAL file for the CRIME variable

In order to view a SpaceStat variable with Idrisi commands such as MAPVAL and HISTO, you must make a .VAL file. The .VAL is used to link a variable to its polygon-ID number. If Idrisi knows where these poly-ID's are in space (i.e. the POLYID image), then the variable in the .VAL file can be mapped by logically relating each data observation to its poly-ID.

1. In order to make .VAL files from SpaceStat, the "Idrisi interface" option must be turned on. Press the **F1** key to go to the options screen, and change the Idrisi interface option (option number **5**) to "YES". Now press **Esc** to return to the regular SpaceStat screen.

2. Use **D-8-3** to go to the "list selected variables" function. Type the dataset filename "**columbus**", and then type the variable names "**id**" and "**crime**" as shown below:

   Variable name: **id**
   Variable name: **crime**
   Variable name: **[Enter]** (press the "Enter" key to finish entering variable names)

3. SpaceStat now asks for the output format. Type "**0**" at the first prompt for the default format. Then press **[Enter]** a couple of times to clear the screen display.

4. The .VAL file should now exist. To verify this, press the **F2** key to escape to DOS, and type:

   ```
   dir   *.val
   ```

   You should see that the file "CRIME.VAL" exists in your work directory. To get back to SpaceStat, type:

   ```
   exit
   ```

   This should bump you back into SpaceStat.

Descriptive statistics from SpaceStat

This procedure calculates some standard statistics for one or more variables in a SpaceStat dataset.

1. Use **E-1-1** to go to the "descriptive stats" function. Choose the "INTERACTIVE" option (option 2). The interactive option means that SpaceStat will prompt you for all needed information. At the prompts, type the following:

problem filename:	**stats.prb**
data set filename:	**columbus**
name for report file:	**[Enter]**
matrix file:	**[Enter]**
variable name:	**crime**
variable name:	**[Enter]**

2. All of the prompts you just entered have been saved in a *problem file* called STATS.PRB. The significance of this will be discussed later. You should now see summary statistics for the CRIME variable displayed on the screen. If any of these statistics are of interest, write them down on a sheet of paper.

When you are done viewing a screenfull of statistics, press **[Enter]** to see the next screen. The final screen reports the results of a Wald normality test on the CRIME variable. This will be covered later in the course--you don't need to be concerned with it now. (If you are interested, the PROB value of 0.54 means that, with a high degree of confidence, the null hypothesis of a normal distribution cannot be rejected.).

3. Pressing **[Enter]** after the last screen takes you back to the SpaceStat menu.

Using the STATS.PRB problem file

4. You can now re-run this procedure without re-typing all of the prompts by using the problem file you made (STATS.PRB). Use **E-1-1** as before, but this time choose the BATCH option (option 1). "Batch" means that SpaceStat expects parameters to come from a file. Enter "**STATS.PRB**" as the problem filename, and you should see the statistics again as in (2) above. Proceed back to the SpaceStat menu by typing **[Enter]** a few times.

   If you want, you can exit to DOS (**[F2]** key), and type "**type stats.prb**" to look at the STATS.PRB file. If you know what you're doing, you can edit this file with a text editor and add or delete variable names, etc. This might be useful later in the quarter.

Mapping a variable with Idrisi (with the MAPVAL program)

You did this in last week's lab, but it is included here again for completeness.

1. If you haven't yet done so, press **[F2]** to go to DOS
2. To map the CRIME.VAL file, type "**mapval crime**".
3. Type "**v**", and add the vector file "COLVEC" with color 8 so that you can see all of the polygon boundaries.
4. Inspect the map. Are there any trends or interesting patterns? Use the cursor (by typing "**c**") to query crime values for individual polyons.

Making a histogram from a .VAL file (with DUMMY and HISTO)

Earlier you used Idrisi's HISTO command to make a histogram for an image of Africa. Making a histogram of such an image is valid, because every grid cell in the Africa dataset corresponds to a data observation. When working with *raster polygons*[1], on the other hand, each data observation can be made up of many grid cells. On the Columbus POLYID image, each neighborhood is composed of at least 20 grid cells. Therefore, a histogram of raster polygons of crime in columbus (150x150 grid cells; 49 polygons), would display values for 22,500 data observations (150^2) instead of the true 49 data observations.

If you want, you can make an incorrect histogram of the Columbus crime values by using the **HISTO** command on the image TEMP1. TEMP1 was created by the MAPVAL program. At the "change min/max values" prompt, enter "**n**". Choose a class width of 5 and graphic output. Compare this histogram to the ones you make in the next section.

---

[1] "Raster polygons" are what we will mostly be using in this course. A raster polygon is a group of grid cells with the same ID value. These grid cells together make up a single polygon (e.g. a county), which is a single data observation.

Dummy image

So we can't make a histogram of every grid cell of the CRIME map. We can, however, "trick" Idrisi by making a *dummy image*. A dummy image contains only one grid cell per data observation, and is meaningless to display in map form. But when used with the **HISTO** command, a dummy image produces a meaningful histogram.

1. To make a dummy image for the CRIME.VAL file, type "**dummy crime**". The DUMMY command inputs a .VAL file and outputs a dummy image of the same name, but with an "X" prefixed to the filename.

2. Now use the **HISTO** command (type "**histo**"). The image to analyze is XCRIME. Don't change the min/max values, and use a class width of 5. Choose graphic output. Note the difference between this histogram and the incorrect one you may have created above.

3. If you want, experiment with different min/max values, different class widths, or with numeric output. Interpret the histograms. What do they tell you about the statistical distribution of crime in Columbus? Are there extreme or unusual values? Is the distribution roughly symmetric?

4. As a final exercise, type **cola xcrime** to display the XCRIME dummy image. You should see 49 raster cells displayed, one for each of the 49 Columbus polygons. As said above, this display is meaningless, however you should be familiar with what a dummy image looks like in case you accidentally display one in the future.

PART TWO - EXTRACTING POLYGON CENTROIDS

This exercise doesn't have anything to do with exploring the CRIME variable. It is included here because you will need to do this on your own dataset. The procedure involves several steps: first, you will be running a program called RASDIST (for RASter DISTance matrix) which computes polygon centroids as well as a distance-based spatial weight matrix). Next you will display these centroids in Idrisi by documenting an Idrisi vector file. Finally, you will be adding the X and Y coordinates for these centroids to the Columbus dataset in SpaceStat. You should still be at the DOS prompt for the next section.

1. Type "**rasdist i**" to run the centroid extraction program (the "i" means you are using an Idrisi image). Enter the name of the image to be processed (POLYID).

2. The next two prompts pertain to the spatial weight matrix that will be created. Since we will not be using this matrix[1], just type "**1**" to both prompts. You will then see the standard output of the RASDIST program.

3. When the program is finished, it will have created three files:

---

[1] We will be using distance-based spatial weight matrices later in the quarter.

centroid.vec:	Idrisi-format vector file
centroid.asc:	ASCII dataset for input to SpaceStat
dmat.asc:	Spatial weight matrix in sparse ASCII format (ignore)

Displaying the centroids in Idrisi

Before we can display the "centroid.vec" file, a *documentation file*[1] must be created, so that Idrisi knows certain parameters about the "centroid.vec" file. The Idrisi command **DOCUMENT** does just this.

1. Type "**document**" to run the **DOCUMENT** command. Select "vector" file type (option 2), and type the vector filename "**centroid**".

2. Idrisi will ask for certain parameters. Accept the default for all of these parameters (7 of them) by pressing **[Enter]** at each prompt.

3. The next screen gives you an overview of the documentation file for "centroid.vec". If you wish, you can press "**1**" and add a descriptive file title, such as "Polygon centroids for Columbus, Ohio". When finished, press **[Enter]**.

4. Now type **col polyid** to display the POLYID image. Type "**v**" to add the vector file "COLVEC" as before, and also add the vector file "CENTROID". You should see the centroid locations on the screen display. If you want, zoom in on part of the image (type "**w**") and re-display the vector files.

Adding the centroids to the SpaceStat dataset

Now we'll add the centroid locations to the existing SpaceStat dataset "columbus". This will take two steps: first, the ASCII file "CENTROID.ASC" will be converted to a SpaceStat dataset (you've done this procedure before). Then, the centroid dataset will be merged with the columbus dataset. The end result will be that the columbus dataset will have two new variables called "X" and "Y". These represent the X and Y coordinates of the centroid points.

1. Type **exit** to return to SpaceStat.

2. Use **D-1-1** to get to the "ASCII to dataset" function. Type "**centroid.asc**" for the ASCII filename, and "**temp**" for the dataset filename. SpaceStat should report that a new dataset called "temp" has been created.

3. Use **D-3-1** to get to the "add variables" function. Type the following:

First data set:	**columbus**
Second data set:	**temp**
Merged data set:	**columbus**

   This will merge the columbus and temp datasets, and overwrite the existing columbus dataset with the new merged columbus dataset. You should now see that the columbus

---

[1] A documentation file for a .VEC file has the extension ".DVC"; for a .IMG file, it is ".DOC". For a .VAL file, it is ".DVL". You will be seeing these filename extensions when you save files to your diskette. Idrisi image, vector, and values files always come in pairs like this: one file contains the data, and one file contains the documentation of the data (a description of vital characteristics).

dataset has the new variables "X" and "Y". These variables are very important and will be used in the next lab to create spatial weight matrices.

HOMEWORK ASSIGNMENT #1

Lab 3 guided you through some analyses with the Columbus, Ohio dataset. Your first homework assignment will be to perform these same analyses on your own dataset. In lab next week, your group should hand in a short writeup on what you learned by exploring **ONE** variable with maps, histograms, and univariate statistics (one writeup per group). The writeup should only be a few paragraphs (ONE PAGE MAXIMUM).

In the writeup, answer and discuss any of these questions if they are pertinent: how is the variable distributed across space? How are the values themselves distributed statistically? Is the distribution symmetric? Are there any extreme values or outliers? Are there any missing data values?

Also in lab next week, your group will be expected to show the TA, on the computer, a map, histogram, and centroid points for your study area.

SETTING UP YOUR OWN DATA IN THE D:\TEMP DIRECTORY

When you work on your own data, you will need certain files in your work directory D:\TEMP. After making sure that the computer is configured for SpaceStat, type:

```
172load   mydata
```

This will copy the files needed for running SpaceStat and Idrisi. Next, you need to copy your data from your diskette to the D:\TEMP directory. This example assumes that your study area is called "FLORIDA". Type:

```
copy   a:\florida.*  d:\temp
```

This should copy three files (.ASC, .IMG, and .DOC). Now, the last thing you need to do is rename the Idrisi image "FLORIDA" to "POLYID", since certain programs you use assume an image called "POLYID". Use the Idrisi command **MAINT** (for "image MAINTenance") to rename the image "FLORIDA" to "POLYID". The Idrisi prompts for this procedure should be straightforward.

Now you are ready to go. Simply re-do this lab, substituting FLORIDA for COLUMBUS and your chosen variable name for CRIME. Your first step should be to convert the ASCII file FLORIDA.ASC to a SpaceStat dataset (Hint: D-1-1).

MAKING A VECTOR FILE OF YOUR POLYGON BOUNDARIES

In lab 3 you added the COLVEC vector file of polygon boundaries. You will probably want such a file for your own study area.

1. Use the Idrisi command **POLYVEC**. You will be prompted for several items:

output vector file: **[name of your choosing]**
background polygon to ignore? **yes**
identifier of background poly: **0**
process polys only in a values file? **no**

2. Now the **POLYVEC** program will begin making the vector boundaries. This will take several minutes--be patient. The program's progress is displayed on the screen.

3. When the vector file has been created, you should use the Idrisi command "**CONVERT V**" (the "v" stands for "vector file") to convert the vector file you made to binary format. This conversion causes the vector file to take up less disk space and to display much more quickly.

4. Now use the **DOCUMENT** command on the new vector file to change the "object type" (option 4) from "poly" to "line".

5. Display the image of your study area with **COLOR**, and type "**v**" to display your new vector file. Verify that everything looks correct.

FINISHING A LAB SESSION ON YOUR OWN DATA

When you are ready to quit the lab session, you will probably want to save certain files back to your diskette, so you don't have to re-create them. Following the above example, if you turned the FLORIDA.ASC file into a SpaceStat dataset, then you will want to copy the dataset back to your diskette. Type:

```
copy   d:\temp\florida.*  a:
```

and make sure that the files FLORIDA.DAT and FLORIDA.DHT get copied to your diskette. The .DAT and .DHT files together make up a single SpaceStat dataset. You should also save any vector files that you made by typing:

```
copy   d:\temp\*.vec  a:
copy   d:\temp\*.dvc  a:
```

A .VEC and a .DVC file together make up an idrisi vector image.

FINAL MAPPING TIP

If the variable you are exploring has a strange distribution, especially if it has extreme observations, your map display from MAPVAL might not be very informative--it might not use all 16 colors effectively. To remedy this, you can use the **DOCUMENT** command to define new minimum/maximum values for the display.

As an example of why you might want to do this, assume you are mapping a variable of population density called POPDENS. Assume further that for most counties, POPDENS takes on values from 100 to 1,000, but one small urban county has a value of 30,000. When Idrisi displays this variable, it divides the range of the variable (from 100 to 30,000, or a range of 29,900) into 16 classes of equal range (a class width of 29,900/16, or 1,869), and the values for POPDENS are assigned to the appropriate class. So the map of POPDENS will have the extreme county in the highest class (28,131 and above, shaded red) and all other counties will be assigned to the lowest class (0 to 1,869) and shaded black. This map tells you essentially nothing about the spatial distribution of POPDENS except where the highest value occurs.

A remedy for this problem is to set the maximum display value from 30,000 to something like 950. This way the values are stretched over a range of 950 instead of 30,000, resulting in a much more informative map display. Here is how to reset the minimum and/or maximum display value.

0. Look at a histogram of the variable and determine the min/max cutoffs you want to use.
1. Remember that the MAPVAL command always produces an image called TEMP1. This is the image you will modify.
2. Use **DOCUMENT** on the TEMP1 image, and change the min/max values (options 15 and 16) to the new desired values.
3. Re-display the TEMP1 image with the **COLA** command. Do not use the MAPVAL command as that will make a new TEMP1 image directly from the .VAL file, and overwrite any changes you made to the old TEMP1 image.
4. If you want to return to the original min/max values, you can run **DOCUMENT** again and let Idrisi re-calculate them, or you can just run MAPVAL again (see #3 above).

LAB 4: Spatial Weight Matrices[1]

INTRODUCTION

The spatial weight matrix (also known as "W" or the "W-matrix") is a fundamental part of many of the spatial statistics you will be using in this course, including spatial autocorrelation indices and spatial regression models. The purpose of the W-matrix is to provide a formal definition of the spatial arrangement of a set of objects (usually points, lines or polygons). The W-matrix defines this spatial arrangement by stating, for each object, which objects are considered as neighbors. Weight values are then attached to these neighbors in one of two ways: one way uses a yes/no criterion where neighboring objects are assigned a value of 1 and non-neighboring objects are given a value of 0. This produces a *binary W-matrix*. The second method uses weight values which reflect the relative strength of each neighbor. For example, if we assume that objects which are closer together have stronger ties as neighbors, then two objects which are 2 miles apart might be given a spatial weight of 0.50, and two objects which are 4 miles apart would be given a spatial weight of 0.25. This type of matrix is called a *general W-matrix*. In this lab you will be dealing with binary W-matrices only.

In addition to the type of weight stored in a W-matrix (binary vs. general), another characteristic of a W-matrix deals with how neighboring objects are identified. The two primary methods of identifying neighbors are based on contiguity and distance. A *contiguity-based W-matrix* identifies objects as neighbors when they are contiguous, or adjacent, in space. A *distance-based W-matrix* identifies neighboring objects when they are within some specified cutoff distance. Note that these two types of W-matrices, contiguity-based and distance-based, can use either binary or general weight values.

An example will be helpful here: let's assume that our study area consists of 48 polygons representing the continental United States. Assume further that we've created a **binary contiguity matrix** (that is, a contiguity-based W-matrix using binary weights). What will this matrix look like?

Since the dataset has 48 observations, the matrix will be of dimension 48 by 48. Each row of the matrix refers to one state, and each column of the matrix refers to one of 48 possible states that could be neighbors. Since the matrix uses binary weights, an element of the matrix can only take on the values 0 or 1, where 0 means "not neighbors" and 1 means "neighbors".

So, taking California as an example, what will the row of the matrix corresponding to California look like? California's contiguous neighbors are Arizona, Nevada, and Oregon. Thus California's row of the matrix will have 3 ones and 45 zeros. The ones will be in the columns which refer to Ariz., Nev., and Ore., and the zeros will be in all other columns. (By convention, an object is not contiguous to itself, thus the column which refers to Calif. has a zero.)

Your homework assignment is to perform, *on your own data*, all of the procedures outlined in this lab. A complete description of the homework is provided at the end of this lab, along with tips on preparing your data.

---

[1] By Rusty Dodson, NCGIA Santa Barbara, 10/30/92.

HOMEWORKS ARE DUE ONE WEEK FROM TODAY

SETTING UP YOUR WORK DIRECTORY

1. Follow your computer lab's procedure for configuring the computer for SpaceStat and Idrisi.

2. You should now be in the D:\TEMP directory. Load data for this lab by typing:

   172load lab4

   This will copy a set of the data you need to the D:\TEMP work directory. Once this data is copied, you are ready to run SpaceStat, Idrisi commands, and various linkage programs.

PART ONE - CONTIGUITY-BASED WEIGHT MATRICES

Extract contiguity from the POLYID image

The Idrisi image POLYID contains all of the information needed to create a binary contiguity matrix. SpaceStat provides a utility for extracting contiguity information from an Idrisi image. In this section of the lab you will create a binary contiguity weight matrix called BIN_1.

1. Run SpaceStat, and use **D-2-3** to go to "Data/GIS Interface/Idrisi format binary contiguity".

2. At the prompt for "matrix filename", type **bin_1**. Enter **polyid** as the Idrisi image name.

   The program will display its progress at extracting the matrix information from the POLYID image. When it is finished, you should see that the matrix BIN_1 was successfully created. (For Columbus, the matrix will be of dimension 49 by 49.)

3. Now you will look at the matrix which was just created. Use **D-8-6** to go to the "List matrix" option. Enter **bin_1** as the matrix filename, **1** for "width" and **0** for "number of decimals". Next you should see the weight matrix displayed row by row (each row "wraps" over more than one line of the screen display). Press **[Enter]** until the entire matrix scrolls by.

Create higher-order W-matrices

Given the basic binary contiguity matrix BIN_1, we will now make higher-order binary contiguity matrices. BIN_1, the first-order matrix, includes as neighbors all polygons which are "one step away", i.e., contiguous polygons. A second-order matrix would include polygons which are up to two steps away, i.e., the first order neighbors plus all polygons contiguous to the first order neighbors. This concept can be extended to higher orders of contiguity.

1. In SpaceStat, use **T-2-2** to go to the "higher order contiguity" option.

2. Enter **bin_1** as the matrix filename, **bin** as the "file name prefix", and **3** as the "highest order of contiguity".

3. You should now have three W-matrices called BIN_1, BIN_2, and BIN_3, which represent first, second and third-order contiguity, respectively.

Row-standardize the W-matrices

In many cases a W-matrix should be standardized in order to control for different polygons having different numbers of neighbors. Row standardization of a W-matrix involves dividing each element of the matrix by the sum of the elements in that row. Thus, on the row-standardized matrix, the sum of the elements on any row is equal to one.

Using the 48 states example, consider Washington state, with its two neighbors Idaho and Oregon. Now consider Tennessee, with eight neighbors[1]. On an unstandardized matrix, Washington would have two ones in its row while Tennessee would have eight ones, indicating that there is four times as much interaction between Tennessee and its neighbors as there is between Wash. and its neighbors. But is this really true? Probably not. More likely, Tennessee has more neighbors due to its elongated shape and its location in the Eastern part of the country, which tends to have smaller states.

A better assumption might be that each state has the same amount of potential interaction with its neighbors, and to divide this interaction among the neighbors. Row standardization achieves this by giving Washington's two neighbors a weight of 0.5 each, and by giving Tennessee's eight neighbors a weight of 0.125 each. Thus, for any state, the sum of the weights for all neighbors equals one.

1. In SpaceStat, use **T-2-1** to go to "Row standardization...".

2. Enter **bin_1** as the matrix filename, and **bin_1s** as the row-standardized filename.

   Use **D-8-6** to go to the "List matrix" option as you did earler. Enter **bin_1s** as the matrix filename, **3** for "width" and **2** for "number of decimals". Note how BIN_1S differs from BIN_1.

3. Repeat step 2 above for the BIN_2 and BIN_3 matrices to make new matrices called BIN_2S and BIN_3S (you don't have to use "List matrix" on these matrices).

Examine the connectivity of W-matrices

Now the six W-matrices you made will be compared by examining summary statistics on the connectivity of each matrix. These statistics pertain to the number of links in the matrix (a link means a nonzero element of the matrix) and to the nature of the weight values stored in these links.

1. In SpaceStat, use **T-5-2** to go to "Connectivity...".

2. At the prompt, enter the matrix filename **bin_1**. You will then be presented with some statistics on the connectivity of the weights matrix. Write down the values of the statistics listed in the table below:

---

[1] Arkansas, Missouri, Kentucky, Virginia, North Carolina, Georgia, Alabama, and Mississippi.

NOTE: Below, "*Average weight*" means the average of all nonzero weights. "*Average #links*" means the average number of neighbors per polygon.

CONNECTIVITY STATISTICS FOR CONTIGUITY-BASED W-MATRICES

Statistic	**BIN_1**	**BIN_2**	**BIN_3**	**BIN_1S**	**BIN_2S**	**BIN_3S**
# nonzero links	_____	_____	_____	_____	_____	_____
% nonzero weights	_____	_____	_____	_____	_____	_____
Average weight	_____	_____	_____	_____	_____	_____
Average # links	_____	_____	_____	_____	_____	_____

3. Repeat step 2 above in order to fill the above table with connectivity statistics for all of the W-matrices.

PART TWO - DISTANCE-BASED WEIGHT MATRICES

This part assumes that polygon centroids were created and are contained in the SpaceStat dataset as variables called "X" and "Y". This should have been done in Lab #3.

In this section you will create two distance-based W-matrices. While the contiguity-based matrices assigned all adjacent polygons as neighbors, the distance-based matrices will assign all polygons within a given cutoff distance as neighbors. Distances are measured as Euclidean distance between polygon centroids. The cutoff distance can be visualized as the radius of a circle drawn around each centroid. Other centroids falling within this circle are included as neighbors and all other centroids are not considered neighbors.

The first step in making the distance-based W-matrices is to create a distance matrix. This matrix stores the distance from each polygon centroid to every other polygon centroid. Once this distance matrix is computed, it is used in creating the W-matrices.

Create a distance matrix

1. In SpaceStat, use **T-4-3** to go to "Create euclidean distance matrix".

2. Enter **columbus**, **dist**, **x**, and **y** in response to the prompts for "dataset filename", "distance matrix", "x-coordinate", and "y-coordinate". This means that you will use the variables X and Y in the COLUMBUS dataset to produce a distance matrix called DIST.

3. Use **T-4-5** to go to "Characteristics of a distance matrix".

4. Enter the name for the distance matrix you just created, **dist**.

   This screen will give you summary statistics for the set of distances between polygon centroids in the dataset. These distances are reported in units of grid cells. For

Columbus, these are arbitrary digitizer units. For other Idrisi images, you can convert grid cell distances to real-world distances by multiplying these distances by the cell size of your POLYID image. For example, if the average distance for your distance matrix is 10 grid cells, and your grid cell size is 2 kilometers, then the average distance between your polygons is 20 kilometers, or about 12.5 miles.

Jot down the summary distance statistics in the table below:

SUMMARY STATISTICS FOR THE DISTANCE MATRIX

Average	Max	Min	Range	Quartiles

Create W-matrices for 2 cutoff distances

1. Look at the distance statistics above and decide on two distance cutoffs for your distance-based W-matrices. Make sure that the cutoff distances are large enough so that no polygons are neighborless (these are identified by SpaceStat as "unconnected observations"). (For Columbus, cutoff values of 25 and 50 units will be fine.)

2. Use **T-4-1** to go to "binary weights from a distance matrix".

3. Enter **dist** for "distance matrix filename", and **dist** again for the new filename prefix.

4. The next prompts ask for distance thresholds. These prompts can be tricky--assuming your two distance cutoffs are 25 and 50 units, type the following at the prompts:

   Upper <= : **25**   Lower > : **0**
   Upper <= : **50**   Lower > : **0**
   Upper <= : **[Enter]**

   Now you will have two W-matrices called DIST_1 and DIST_2, representing the cutoff distances of 25 and 50 units, respectively.

Examine and compare connectivity of W-matrices

1. In SpaceStat, use **T-5-2** to go to "Connectivity...".

2. Obtain connectivity statistics for the matrices DIST_1 and DIST_2. Fill in the table below:

CONNECTIVITY STATISTICS FOR DISTANCE-BASED W-MATRICES

Statistic	DIST_1	DIST_2
# nonzero links	_____	_____
% nonzero weights	_____	_____
Average weight	_____	_____
Average # links	_____	_____

PART THREE - OPTIONAL ADDITIONAL WORK

One application of a W-matrix is to create a *spatially-lagged variable*, also known as a "spatial lag variable" or "lag variable". A lag variable is produced by using matrix algebra to multiply a W-matrix and a variable. If the W-matrix is row-standardized, then the resulting spatial lag variable represents the weighted average of all neighboring values. In this exercise, you will make two spatially-lagged variables from the CRIME variable in the Columbus dataset.

Create spatially-lagged variables

1. In SpaceStat, use **T-1-1** to go to "Spatial lag".
2. Enter the dataset name **columbus** and the spatial weights filename **bin_1s**.
3. You will be prompted for the name of an existing variable, and for a new variable which will be created. Type **crime** for "existing variable" and **w_c1** for "new variable". Then press **[Enter]**. You should now see that the new variable W_C1 was added.
4. Do steps 1-3 above, using the spatial weights matrix **bin_2s** and the new variable name **w_c2**.

Make .VAL files for the new variables

1. Press **F1** and set the "idrisi interface" option to "yes".
2. Use **D-8-3** to go to "Listing selected variables...".
3. Type **columbus** as the dataset, and then type the variable names **id**, **crime**, **w_c1**, and **w_c2**. Press **[Enter]**, and then type **0** to accept the default output format.

Map the new variables

1. Quit SpaceStat (**Alt+q**), and use MAPVAL to display the values files CRIME, W_C1, and W_C2.

2. Observe how the spatially-lagged crime variables are different than the original crime variable. If you wish, make histograms of the new variables (using DUMMY and HISTO).

3. A final thing to try is to use Idrisi's OVERLAY command to overlay images of CRIME and W_C1. Use the "subtract" option and interpret what the new image represents. (To make images of the variables, remember that MAPVAL makes an image called TEMP1. You can use MAINT to rename the TEMP1 image to some other name, and then use MAPVAL again to make an image of the next variable.)

   An alternative to overlaying in Idrisi is to use **D-6-2** to make a new variable which is (CRIME - W_C1). Then make a .VAL file for the new variable and map it in Idrisi.

HOMEWORK ASSIGNMENT #2

As said before, the homework assignment is to perform the procedures described in this lab on your own data. In addition, each group should hand in written answers to the following questions. Hand in your answers on a separate sheet of paper.

1. Look at the connectivity statistics for the contiguity-based matrices BIN_1, BIN_2, BIN_3, BIN_1S, BIN_2S, and BIN_3S.

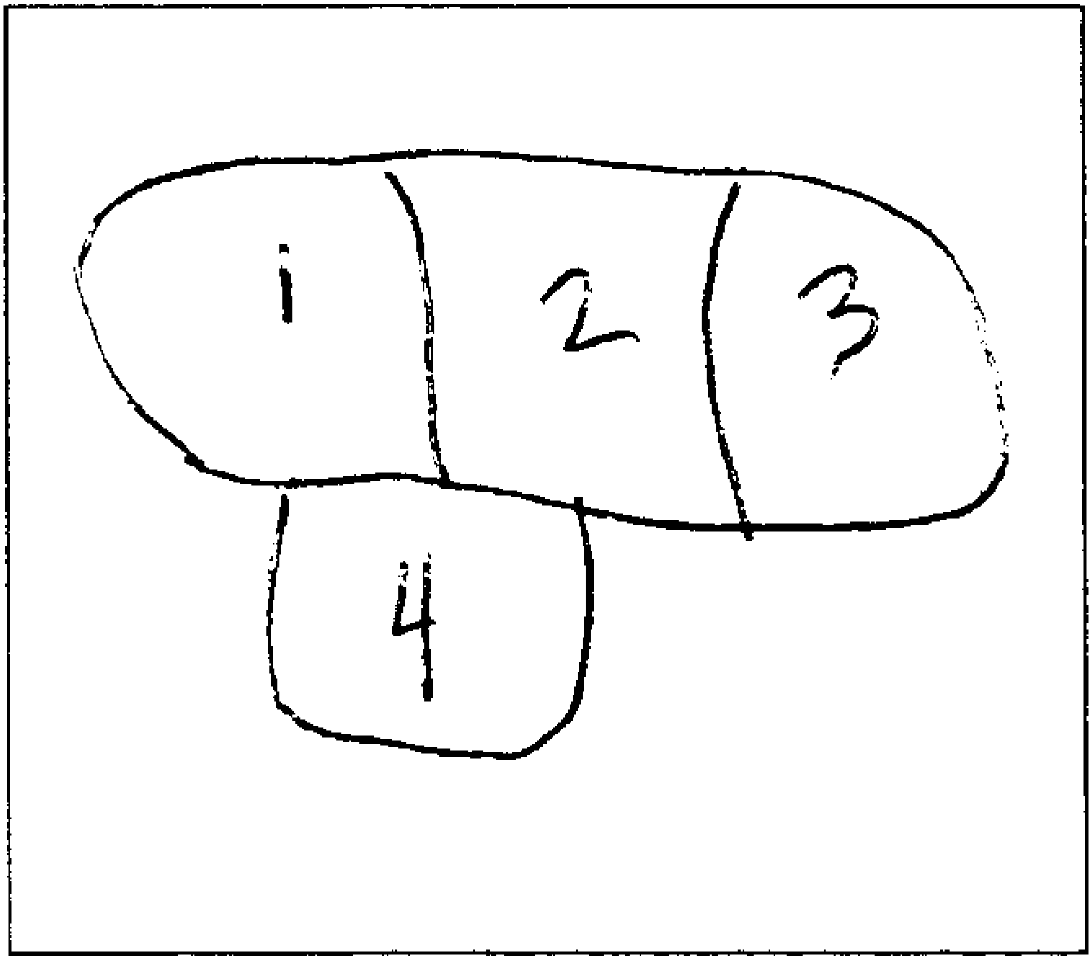
   a) As we move from first-order to higher-order contiguity, which of the connectivity statistics change and why? Which stay the same and why?

   b) Now looking at the row-standardized vs. the unstandardized matrices, which statistics change and why? Which stay the same and why?

2. Looking at the distance-based matrices DIST_1 and DIST_2, how does the distance cutoff affect the connectivity of the matrix? What would the matrix look like if the cutoff distance were greater than the maximum distance between polygons?

3. Look at the connectivity statistics for the matrices DIST_1, DIST_2, BIN_1, and BIN_2. In general, how does matrix connectivity compare between the contiguity-based and the distance-based W-matrices?

4. Create (on paper) two W-matrices for the data set of four polygons shown below. One matrix should use binary contiguity, and the second matrix should be a row-standardized version of the first.


1
2
3
4

SETTING UP YOUR OWN DATA IN THE D:\TEMP DIRECTORY

When you work on your own data, you will need certain files in your work directory D:\TEMP. After making sure that the computer is configured for SpaceStat, type:

```
172load   mydata
```

This will copy the files needed for running SpaceStat and Idrisi. Next, you need to copy your data from your diskette to the D:\TEMP directory. The following command copies all files from your diskette to the work directory D:\TEMP.

```
copy   a:\*.*  d:\temp
```

Remember that your image of polygon-IDs needs to be named POLYID.

FINISHING A LAB SESSION ON YOUR OWN DATA

If you want to save the matrices you created for this lab, typing

```
copy   *.fmt   a:
```

will copy all SpaceStat matrices to your diskette. (SpaceStat uses the .FMT extension for all matrices.)

LAB 5: Point Pattern Analysis[1]

INTRODUCTION

This lab provides quick, interactive experience with some spatial statistics for point patterns. In the first part of the lab, you will run a stand-alone program and create your own arbitrary point patterns. In the second part you will work with some real data.

The PCENTER program was written by Professor Peter F. Fisher of Kent University, Ohio, and was obtained though the public domain listserver (VMSSERV@LEICESTER.AC.UK) maintained by the Computers in Teaching Initiative Centre for Geography (CTICG).

The London Cholera data were digitized by Rusty Dodson from the map included in the book by John Snow, "Snow on cholera...", London, Oxford University Press, 1936.

SETTING UP YOUR WORK DIRECTORY

1. Follow your computer lab's procedure for configuring the computer for SpaceStat and Idrisi.

2. You should now be in the D:\TEMP directory. Load data for this lab by typing:

    172load lab5

    This will copy a set of the data you need to the D:\TEMP work directory.

PART ONE - THE "PCENTER" PROGRAM

In this part of the lab you will use a stand-alone program called PCENTER to explore various distance-based spatial statistics. In class you were introduced to the concepts of **nearest neighbor**, **mean center**, **standard distance**, and **standard ellipse**. The PCENTER program allows you to explore these statistics interactively by creating your own point pattern directly on the computer screen. You will use the arrow keys on the keyboard to move a cursor and place points on the screen. At any time, you can press **S** to calculate statistics based on the current point pattern you have created. The computer will display the nearest neighbor statistic on the screen, and tell you if the pattern is "more even" or "more clustered" than normal. In addition, the standard circle and standard ellipse will be drawn on the graphics screen so you can visualize these statistics.

Running the PCENTER program

1. At the D:\TEMP> prompt, type **pcenter**.

2. Press **K** to indicate that you want to enter points directly from the keyboard.

3. Now use the arrow keys of the keyboard to move the cursor around. When you want to place a point, press the **Insert** key.

---

[1] By Rusty Dodson, NCGIA Santa Barbara, November 6, 1992.

4. After placing several points, press **S** to calculate and display statistics. The nearest neighbor statistic will appear in the upper-right screen corner, along with the message "more even" or "more clustered". This message describes the point pattern with respect to a random point pattern.

   On the graphics screen, you will also see the location of the mean center, the standard circle, and the standard ellipse which describe the current point pattern. (The rectangle you see represents the minimum bounding box for your points--this is the box which is used in computing the density of your point pattern.)

5. Continue to add points to the point set and re-calculate the statistics at any time by pressing **S**. Note how the positions of the mean center and standard circle and ellipse change. Also notice how the nearest neighbor statistic changes between "more even" and "more clustered".

6. If you want to start over with a blank screen, press **X** to exit, and then press **K** to go back to point-entry-mode. Experiment with different point patterns until you get bored.

Redefining the study area

In order to calculate the significance of the nearest neighbor statistic (i.e. to tell whether the point pattern is "even" or "clustered"), you need to know the area of the study area. In the exercise above, the PCENTER program used the minimum bounding box as the study area boundary. In this section you will re-define the study area by drawing a box or polygon around your point set. If you start with a point pattern which is "more even", you may find that re-defining the study area causes the same point set to become "more clustered". This is a classic example of how spatial statistics can be sensitive to conditions at the boundary of the study area. Instructions for drawing the new study area boundary follow:

1. Press **O** to "draw outline".

2. Move the cursor with the arrow keys, and press **Insert** to draw a vertex of the boundary polygon.

3. Trace around the desired polygon boundary, placing vertices with the **Insert** key.

4. When the polygon is drawn, press **P** to "return to point mode and snap". This will close up the polygon and return you to point-entry-mode.

5. Press **S** to compute new statistics for the exact same point pattern, but based on the new study area boundary. Did re-defining the boundary affect the "even" or "clustered" description of the point set?

6. When finished, press **X** twice to exit the program.

PART TWO - LOOKING AT REAL POINT DATA WITH IDRISI

London cholera data

This famous point set pertains to an epidemic of cholera in London during the year 1854. Each point is associated with the residence of a person who died from cholera. At the time of this epidemic, the cause of cholera was unknown. A local physician in the area named John Snow mapped the incidences of death and examined the spatial pattern of the map. A simple visualization of the data led Dr. Snow to a conclusion about the way cholera is transmitted. You will learn more about this later. In this part of the lab you will use Idrisi to calculate the mean center of the cholera death points.

1. First, type **col deathras** to display a raster image of the death points called DEATHRAS. (The program COL is a shortcut to Idrisi's COLOR command. It saves you from answering all of Idrisi's questions.)

2. Type **v** and add the vector file **street**. Use color **7**. This will add the London street network to the image display.

Now that you've seen what the data look like, you will now calculate the mean center of the points. First press **Esc** to clear the screen display.

1. Type **center** to run the Idrisi command that computes the mean center. Enter the image name **deathras**.

2. On the screen you will see statistics about the point pattern. Write down the column/row position of the mean center, and write down the length of the standard radius in cell widths.

   Mean center:
   column = _______
   row = _______
   standard radius in cell widths = _______

3. Now redisplay the point data with Idrisi. While Idrisi doesn't display the mean center for you, you have been provided with a vector file called CIRCLE that shows the mean center and the standard circle. The following steps create a decent map display:

   a) Type **col deathras**

   b) Type **v** and add the vector file **street** with color **7**.

   c) Type **v** and add the vector file **circle** with color **5**. The CIRCLE file shows the standard circle and mean center location for the cholera point pattern. Press **c** and verify that the position of the mean center is correct.

   Don't erase this display until you read the next section.

Dr. Snow's conclusion

Why are the cholera deaths distributed the way they are? Dr. Snow was trying to answer this question by mapping the deaths, just as you have done. However, the good doctor had one more piece of information that you haven't yet seen. There is another vector file in this dataset called PUMP, which contains the locations of water pumps from which the citizens obtain their drinking water.

1. Type **v** and add the vector file **pump** with color **15**.

At the time of the cholera epidemic, some researchers speculated that cholera was transmitted through the air, while a smaller group suspected that the disease was water-borne. One researcher went through an elaborate analysis in which he showed a strong negative correlation between elevation and incidences of cholera, postulating some sort of interaction between the environment and the cholera disease.

Based on the pattern of cholera deaths in relation to the locations of water pumps, Dr. Snow theorized that cholera was transmitted through water, and that the epidemic was caused by a particular water pump being contaminated with the phenomena which causes cholera. (Guess which pump was the culprit.)

Yes, indeed, the pump that is right next to the mean center[1] was the one suspected by Dr. Snow, who took corrective action by removing the pump handle. This prevented the townsfolk from drinking the contaminated water, and led to an enormous drop in cholera deaths within the next few days.

Issues to think about

1. Given that the people in London had to travel along the street network in order to get to a water pump, what are the implications of using Euclidean distance in computing the mean center and standard circle? What might be a better way to measure distance?

2. In computing the mean center, what assumptions are made regarding the underlying population distribution of London? Do you think this is likely to affect the analysis results?

---

[1] This pump is on Broad Street, and is infamously known as the "Broad Street pump".

LAB 6: Spatial autocorrelation statistics[1]

INTRODUCTION

Spatial autocorrelation statistics measure the degree of clustering exhibited by a variable over space. In this lab you will compute and interpret two types of spatial autocorrelation statistics: the Moran's I statistic and the Getis-Ord G statistics. These statistics operate on interval and ratio variables. Moran's I works on any interval/ratio variable, while the G statistics require that the variable be non-negative as well. Both statistics require a spatial weights matrix (W-matrix) in order to be computed. Moran's I can be used with any W-matrix, while the G statistics should only be used with distance-based W-matrices[2].

Important:

This lab is written to work with the COLUMBUS dataset, on the variable CRIME. The lab also assumes the presence of **four** spatial weight matrices: BIN1, D30, D45, D60, and D75. BIN1 is a binary contiguity matrix. D30, etc. are binary distance-based matrices, using increasing distance cutoffs of 30, 45, 60, and 75 units. **ALL MATRICES ARE ROW-STANDARDIZED**.

When you run this lab on your own data, you will need W-matrices corresponding to those described above. You should have a row-standardized binary contiguity matrix left over from lab 4 (it would be called BIN_1S.FMT). If you don't have it, refer to lab 4 for instructions on re-creating it.

For the distance-based matrices, you will obviously have to pick your own cutoff distances (four of them) rather than using the same numbers in the Columbus matrices above. Again, the procedure for making these matrices is explicitly outlined in lab 4. Here's a quick review: use **T-4-3** to make a distance matrix; use **T-4-1** to make W-matrices based on the distance matrix; use **T-2-1** to row-standardize a W-matrix.

When you use your own data, make the appropriate substitutions for dataset filename, variable name, and spatial weights filenames.

THE HOMEWORK ASSIGNMENT IS TO HAND IN ANSWERS TO THE SIX QUESTIONS (A,B,C,D,E,F) WHICH APPEAR THROUGHOUT THIS DOCUMENT.
HOMEWORKS ARE DUE ONE WEEK FROM TODAY

SETTING UP YOUR WORK DIRECTORY

1. Follow your computer lab's procedure for configuring the computer for SpaceStat and Idrisi.

2. You should now be in the D:\TEMP directory. Load data for this lab by typing:

---

[1] By Rusty Dodson, NCGIA Santa Barbara, November 13, 1992.

[2] The G-stats also require that the matrix be binary and not row-standardized, but SpaceStat takes care of this automatically, as long as the W-matrix is based on distance, rather than contiguity.

```
172load lab6
```

This will copy a set of the data you need to the D:\TEMP work directory. Once this data is copied, you are ready to run SpaceStat, Idrisi commands, and various linkage programs.

PART ONE - MORAN'S I

This section explores the spatial autocorrelation of the CRIME variable through the Moran's I statistic. Our null hypothesis throughout this lab will be that the CRIME variable is randomly-distributed across space, i.e. that there is no significant spatial autocorrelation. A significant value for the I statistic will lead us to reject the null hypothesis, and accept the alternative hypothesis of either positive spatial autocorrelation (more clustered than random) or negative spatial autocorrelation (more dispersed than random). The sign of the I statistic determines whether positive or negative autocorrelation is detected.

In order to assess the significance of the I statistic, three possibilities exist in SpaceStat. The first two involve making assumptions about the variable in order to mathematically compute a standard Z-value for I. The final option is based on an empirical permutation approach. The three options for computing Moran's I significance in SpaceStat are:

1. The *normal assumption*, in which the variable in question is assumed to be normally-distributed. Note that you can test a variable for normality in SpaceStat with the **E-1-1-2** function.

2. The *randomization assumption*, in which one assumes that each observation is equally likely to occur at any location in space.

3. The *permutation approach*, where a large number of values for I are computed based on random permutations of the data.

In options 1 and 2 above, a standard Z-value for I is computed mathematically. From the Z-value, a significance probability is obtained, based on the standard normal distribution. In option 3 above, a Z-value is NOT computed. Instead, a *pseudo-significance probability* is obtained based on the empirical values of I from the random data permutations.

Moran's I - normal assumption

Making a problem file

For the first Moran's I option in SpaceStat (the normal assumption), you will make a *problem file*. This problem file can then be used for the other options (randomization and permutation), so you won't have to re-type the same parameters.

1. Type **space** to run SpaceStat.

2. Use **E-3-1-2** to go to the Explore/Moran/Normal/Interactive function. "Interactive" means you will be typing in all the parameters from the keyboard.

3. Now you will be prompted for several items. Type in the following at the prompts:

   ```
   Problem filename:        moran.prb
   Data set filename:       columbus
   ```

Report file:	**[Enter]**
Matrix file:	**bin1**
Matrix file:	**[Enter]**
Variable name:	**crime**
Variable name:	**[Enter]**

4. After a short wait, you should see a verification that the W-matrix is row-standardized. Press **[Enter]**, and you will then see the output from SpaceStat's computation of Moran's I. Write down the following values:

I-stat	**Z-value**	**Prob.**
_____	_____	_____

HOMEWORK QUESTION A

A) Is the I statistic for your variable significant? Explain. Include, in your answer, the appropriate SpaceStat output as in #4 above.

Moran's I - randomization assumption

Running a problem file ("batch" mode)

Now you will compute Moran's I for CRIME, but this time using the randomization approach for calculating significance. You will use the problem file you created above.

1. Use **E-3-2-1** to go to Explore/Moran/Randomization/Batch. "Batch" means SpaceStat will be taking its input from a problem file (this is called "batch mode" in computer jargon).

2. Type the problem file name, **moran.prb**. (Remember the ".prb" extension.)

3. After a short wait, press **[Enter]** and you should see the output from SpaceStat's computation of Moran's I. Write down the following values:

I-stat	**Z-value**	**Prob.**
_____	_____	_____

HOMEWORK QUESTION B

B) What values have changed, compared to the normal assumption? Why? Include, in your answer, the appropriate SpaceStat output as in #3 above.

Moran's I - permutation approach

Now you will again compute Moran's I, this time using the permutation approach.

1. Use **E-3-3-1** to go to Explore/Moran/Permutation/Batch.

2. Type the problem file name, **moran.prb**. (Remember the ".prb" extension.)

3. After a short wait, press **[Enter]** and you should see the output from SpaceStat's computation of Moran's I. Write down the following values:

I-stat	Prob.
_____	_____

HOMEWORK QUESTION C

C) Why is the significance probability so different here as compared to the normal and randomization assumptions? Using the permutation approach, how would you determine whether the I statistic was significant at the 0.001 level? Include, in your answer, the appropriate SpaceStat output as in #3 above.

Moran's I - spatial correlogram

The *correlogram* option in SpaceStat is used for investigating the changes in spatial autocorrelation when it is computed using different W-matrices. Since the computation of these statistics depends on the W-matrix, and the W-matrix is defined somewhat subjectively by the analyst (you), it follows that any statistical output is dependent on and sensitive to the form of the W-matrix. Thus it is always a good idea to compute spatial statistics using several different W-matrices. This procedure is termed *sensitivity analysis*. A sensitivity analysis is used to assess the stability of a statistic or model to changes in its inputs. In this case, the input we are concerned with is the W-matrix.

This section walks you through a computation of a Moran's I correlogram for the normal assumption only. You are free to explore the other options.

1. Use **E-3-4-2** to go to Explore/Moran/Correlogram-Normal/Interactive.

   (Remember, use "interactive" when you need to *create* a problem file. Use "batch" when you want to *use* a pre-existing problem file.)

2. Now you will be prompted for several items. Type in the following at the prompts:

   Problem filename: **corgram.prb**
   Data set filename: **columbus**
   Report file: **[Enter]**
   Matrix file: **d30**
   Matrix file: **d45**
   Matrix file: **d60**
   Matrix file: **d75**
   Matrix file: **[Enter]**
   Variable name: **crime**
   Variable name: **[Enter]**

3. After a short wait, press **[Enter]** and you should see the output from SpaceStat's computation of Moran's I. Write down the following values:

W-matrix	I-stat	Z-value	Prob.
d30	_____	_____	_____
d45	_____	_____	_____

d60	_____	_____	_____
d75	_____	_____	_____

HOMEWORK QUESTION D

D) How does the I statistic change with different weight matrices? What does this mean? Include, in your answer, the appropriate SpaceStat output as in #3 above.

PART TWO - THE G STATISTICS

The Getis-Ord G statistic family is similar in some respects to the Moran statistic. However, it has important differences. First, the G-statistics detect only *positive spatial autocorrelation* (clustering). The sign of the G statistics determines whether the autocorrelation is dominated by a clustering of high values (positive G) or a clustering of low values (negative G). Note that the sign of the statistic is reflected by the sign of its Z-value.

A further distinction between I and G is that while Moran's I is a global statistic only, the G family includes both a global and a local measure of autocorrelation. A *global statistic* describes a distribution with a single number, while a *local statistic* provides a numerical measure for each observation in a distribution (for a total of N numbers).

The global G statistic

Compute a correlogram using the corgram.prb file created earlier.

1. Use **E-5-2-1** to go to Explore/G-stats/correlogram/batch.

2. Type the name of the problem file, **corgram.prb**.

3. After a short wait, the computer will warn you that it is treating the row-standardized W-matrices as binary (unstandardized). Press **[Enter]** and you should see the correlogram for the G statistic, using four distance-based W-matrices. Write down the following values:

W-matrix	G-stat	Z-value	Prob.
d30	_____	_____	_____
d45	_____	_____	_____
d60	_____	_____	_____
d75	_____	_____	_____

HOMEWORK QUESTION D

D) How does the G statistic change with different W-matrices? Are the various G-stats significant? What can the G statistic tell you that Moran's I can not? What can Moran's I tell you that G can not? Include, in your answer, the appropriate SpaceStat output as in #3 above.

Local G statistics: the Gi* statistics

The Gi* statistics measure clustering for each observation of a variable. (The "i" means "for all *i* observations".) An individual Gi* measure indicates whether an observation *i* plus its neighbors constitute a cluster of high values (positive Z's) or a cluster of low values (negative Z's). Since each observation has a value, the Gi* statistics can be displayed as a map. In this section, you will calculate the Gi* statistics and create .VAL files for the statistics which will be used to create maps in Idrisi.

1. In order to make the necessary .VAL files, you will need to set some SpaceStat options. Press **F1** to go to the options screen, and set option #4 (**Indicator variable**) to **ID**, the name of your polygon-ID variable. Also, set option #5 (**Idrisi interface**) to **yes**.

   NOTE: Option 4 (Indicator variable) is critical when creating .VAL files for Gi* statistics. If you don't use it, your map displays might be wrong.

2. Use **E-5-4-2** to go to the Explore/G-stats/G-I*-stat/Interactive function.

3. Now you will be prompted for several items. Type in the following at the prompts:

Problem filename:	**gstat.prb**
Data set filename:	**columbus**
Report file:	**[Enter]**
Matrix file:	**d30**
Matrix file:	**[Enter]**
Variable name:	**crime**
Variable name:	**[Enter]**

HOMEWORK QUESTION E

E) The output screen shows you the observations with the 10 largest and smallest Z-values. Which Z-values (positive or negative) tend to be higher in absolute value?. How does this relate to what the global G statistic told you?

Mapping the Gi* statistics

Since you set the appropriate options ("Indicator variable" and "Idrisi interface"), SpaceStat has created three values files from the output of the Gi* statistics. These files are CRIME_G.VAL, CRIME_P.VAL, and CRIME_Z.VAL, and represent the values for the Gi* statistics, the significance probabilities, and the Z-values, respectively. In this part of the lab you will look at two maps: one of the values for the Gi* statistics themselves, and one which only displays the statistically significant Gi* statistics.

1. Type **Alt+Q** to quit SpaceStat.

2. Type **dir *.val** to determine that the .VAL files were created. If not, check the procedure of the previous section to make sure you set the SpaceStat options correctly.

3. First we will look at the Gi* statistics themselves. Type **mapval crime_g**. Add the vector file COLVEC with color 0 to make the display more interpretable. Press **c** and use the cursor to look at the individual Gi* values for the polygons.

The map of the Gi* values is interesting, but it doesn't tell us anything about which values are statistically significant. Therefore, we will make a map of the Z-values which highlights only the polygons with significant Gi* values. We use the Z-values for this map because the Z-values can be converted directly to significance probabilities. For example, based on the normal probability table, a Z-value greater than +/-1.96 corresponds to a significance probability of 0.05 (a 95% confidence interval). The following map displays the significant Z-values according to their degree of significance. Polygons with significant positive Z-values (which correspond to clusters of *high* values) are displayed with shades of red, and polygons with significant negative Z-values (clusters of *low* values) are displayed with shades of blue. The brighter the color, the more significant the associated Gi* statistic. Polygons with non-significant Gi* statistics are shaded gray. The program which shades Z-values based on significance probability is called GZ.

4. Type **gz crime_z** to make your map of significant Gi* statistics. Add the vector boundary file for interpretability. What does this map indicate?

PART THREE - OPTIONAL ADDITIONAL WORK

Making Gi* statistics for additional distance cutoffs

Rename the CRIME_Z.VAL file to CR30_Z.VAL, in order to be able to identify which distance cutoff goes with this values file. (In DOS, you would type "rename crime_z.val cr30_z.val" to rename the file.)

Now restart SpaceStat, and re-do the section above called "Local G statistics: the Gi* statistics". This time, specify a W-matrix with a different distance cutoff. Remember to set the SpaceStat options so that you make a new values file called CRIME_Z.VAL. That's another reason you renamed the old CRIME_Z.VAL file--so that the new one didn't write over the old one. If you wish, rename this CRIME_Z.VAL file, and make another set of Gi* statistics based on yet another critical distance.

Now quit SpaceStat, and use GZ to display your 2 or 3 values files for of Gi* statistics. Note how the significant polygons change with the changing cutoff distance.

SETTING UP YOUR OWN DATA IN THE D:\TEMP DIRECTORY

When you work on your own data, you will need certain files in your work directory D:\TEMP. After making sure that the computer is configured for SpaceStat, type:

```
172load   mydata
```

This will copy the files needed for running SpaceStat and Idrisi. Next, you need to copy your data from your diskette to the D:\TEMP directory.

Now make sure you copy, from your diskette, your SpaceStat dataset (.DAT and .DHT files), your POLYID image (both of its parts, .IMG and .DOC), your vector boundary files (.VEC and .DVC), and any other important files (values files, dummy images, etc.).

FINISHING A LAB SESSION ON YOUR OWN DATA

When you are ready to quit the lab session, remember to copy newly-created files back to your diskette.

LAB 7: Regression Analysis[1]

INTRODUCTION

Regression analysis is concerned with finding a relationship between a dependent variable and one or more explanatory variables. In this lab you will use SpaceStat to compute several regression models using the ordinary least squares (OLS) estimator. The OLS estimator finds a linear relationship between the dependent and explanatory variables such that the sum of the squared errors is minimized. The equation for a bivariate[2] regression model estimated from a sample is:

$$\mathbf{Y = a + bX + e}$$

Where **Y** refers to N observations of a dependent variable and **X** refers to N observations of an explanatory variable. These are known values. The parameters **a**, **b**, and **e** are are estimated from the regression model. **a** refers to the constant term (the Y-intercept of the regression line), and **b** is the coefficient of the explanatory variable (the slope of the regression line). These are single values. **e** refers to a set of N residuals (one per observation). The residuals are the part of the model that is not explained by the explanatory variable.

This lab guides you through three types of regression analysis: standard bivariate regression, spatial analysis of variance (spatial ANOVA), and trend surface regression. The lab is written for the Columbus, Ohio dataset. You are free to substitute your own dataset.

SETTING UP YOUR WORK DIRECTORY

1. Follow your computer lab's procedure for configuring the computer for SpaceStat and Idrisi.

2. You should now be in the D:\TEMP directory. Load data for this lab by typing:

```
172load lab7
```

This will copy a set of the data you need to the D:\TEMP work directory. Once this data is copied, you are ready to run SpaceStat, Idrisi commands, and various linkage programs.

PART ONE - ORDINARY BIVARIATE REGRESSION

Our objective in this part of the lab is to build a regression model which uses an explanatory variable to explain a dependent variable. Suppose our dependent variable--the variable we want to model, or explain--is the CRIME variable from Columbus. Suppose further that we have two additional variables, INCOME (average household income) and HOUSING (average housing value), and that we hypothesize that crime is related in some way to both of these variables. Our first step might be to explore these variables graphically in order to decide whether we want to use INCOME or HOUSING as the explanatory variable.

---

[1] By Rusty Dodson, NCGIA Santa Barbara, November 29, 1992.

[2] *Bivariate* means "using two variables". A bivariate regression has one dependent and one explanatory variable.

Displaying a scatterplot of two variables

A *scatterplot* is a graphical tool for exploring the relationship between two variables. Scatterplots are often used as a preliminary "quick look" at some data, before a formal regression model is estimated. You will be using an Idrisi module called REGRESS to make scatterplots. The REGRESS module also computes a bivariate regression, but this will be mostly ignored since you will be using SpaceStat to perform regressions.

1. Make sure that values files exist for all variables in which you are interested. For the Columbus data, you should have the files CRIME.VAL, INCOME.VAL, and HOUSING.VAL.

2. Now type **regress** to start the REGRESS module. Choose option **1**, since you are working with values files. Next you will see several prompts:

   Name of values file with independent variable: **income**
   Name of values file with dependent variable: **crime**

NOTE: The REGRESS module will probably not work if SpaceStat is in memory. If you get an "Insufficient memory..." error, quit SpaceStat and try REGRESS again. If it still doesn't work, use DUMMY to make dummy images out of your values files and run REGRESS on the dummy images using the "image" option (option #2).

Now you should see the scatterplot displayed on the screen. The dependent variable, CRIME, is mapped on the vertical axis and the explanatory variable, INCOME is mapped on the horizontal axis. In addition, the OLS regression line is displayed on the plot, and the regression equation is included at the top of the screen. Press Esc and you will see a screen of additional statistics, including the correlation coefficient (r) and the (r2) goodness-of-fit measure.

* What type of relationship (strong/weak/nonexistent, linear/nonlinear, positive/negative, homoskedastic/heteroskedastic) do you see between crime and income in Columbus?

* Look at a scatterplot of crime vs. housing. What type of relationship is this? Which pair of variables has the stronger relationship? What criteria are you using to assess the strength of these relationships?

The regression model

From looking at the scatterplots, you probably guessed that INCOME has a stronger relationship to CRIME than HOUSING does. Now you will use SpaceStat to estimate a regression, using the INCOME variable to explain CRIME.

1. Type **space** to run SpaceStat.

2. Press **F1** to set the following options:

   Option 4: Indicator variable (**ID**)[1]
   Option 5: Idrisi interface (**YES**)

---

[1] If you don't set the indicator variable to "ID", the poly-ID variable, then your Idrisi maps will be wrong because SpaceStat will assume your polygons are numbered continuously starting from 1. All of the county datasets start numbering at 2, and some have discontinuities.

3. Use **R-1-1-2** to go to Regress/Classic/OLS/Interactive.

4. Type the following at the prompts:

Problem file name: **reg1.prb**
Data set filename: **columbus**
Name for report file: **[Enter]**
Matrix file: **[Enter]**
Enter option number: **1** (generic regression)
Constant term: **[Enter]**
Variable name: **crime** (enter dependent variable first)
Variable name: **income**
Variable name: **[Enter]**
Heteroskedastic variable name: **[Enter]**

After a short wait, you will see the first screen of output. Look at the goodness-of-fit measures **R2** and **R2-adj**, and the **t-value** and significance probability for the INCOME variable.

* How good is the fit of the model? How much of the variance in CRIME is explained by INCOME?

* Is the INCOME variable significant? At what level?

* What is the sign of the coefficient for INCOME? What does this tell you?

The next two screens of output are regression diagnostics. In this course, we will not be looking at these diagnostics in any detail--they will be covered in a later course. Press **[Enter]** to see the second screen of output. This screen shows the results for tests on multicollinearity (correlation among the explanatory variables) and on the distribution of the regression errors. Press **[Enter]** again and you will see results of a test for heteroskedasticity, and a general test of model specification.

Mapping output from the regression

Since you set the appropriate SpaceStat options in step 2 above, you should have two values files which contain the predicted values and the residuals from the regression. These files have the same name as your dependent variable, with a suffix of "_YP.VAL" or "_E.VAL" added. The variable name will be truncated if necessary to make the filename no longer than 8 characters.

For Columbus, you should have the files CRIME_YP.VAL and CRIME_E.VAL. The "YP" stands for "Y-predicted", and the "E" stands for "errors". The CRIME_YP file contains the values for crime when predicted by the INCOME variable using the regression equation:

Predicted crime = a + b(INCOME)

Where **a** and **b** are the parameters estimated by your regression model (constant and coefficient, or, if you prefer, intercept and slope).

The file CRIME_E contains the residuals of the regression. These regression residuals are simply the difference between the actual crime values and the predicted crime values.

Residual = observed value - predicted value

A positive residual indicates that your model is under-predicting the dependent variable. A negative residual denotes over-prediction. A residual with a relatively high absolute value (i.e. a large error) means that the model is not predicting that observation very well. Residual values very close to zero indicate areas where the model is accurately predicting the dependent variable.

To map your regression output, follow these steps:

1. Press **F2** to go to DOS.

2. Type **mapval crime_yp** to map the predicted values for crime. How does the overall pattern of predicted crime values match that of the actual crime values?

3. Type **resids crime_e** to map the residuals.

   NOTE: The residuals can be displayed with MAPVAL, but the display is somewhat messy due to the negative values. The RESIDS function divides the residuals into seven classes of negative values (blue colors) and seven classes of positive values (red colors).

   On the map of residuals, look for extreme values or interesting patterns such as clusters of like-valued errors. These might give you insights into how your model is predicting over space.

PART TWO - SPATIAL ANOVA

A spatial ANOVA regression is used to test whether the mean of a variable is significantly different between two or more spatial subsets of the data. To do this, the ANOVA model uses as its explanatory variable a spatial regime variable. This regime variable is a categorical variable which defines two or more subsets of a study area.

Recall the map of crime in Columbus--higher values generally occur in the center of the study area (the inner city) while lower crime values occur in the outskirts (residential areas). The hypothesis we will test with the ANOVA model is this: is the mean of CRIME significantly different between the center and the periphery of Columbus? Or is the apparent spatial pattern of crime simply due to chance? Before computing a spatial ANOVA, you need a spatial regime variable which reflects the hypothesis you want to test. For Columbus, we will use an existing regime variable called CP (center/periphery). For your own data, you need to come up with a hypothesis and then create a spatial regime variable with which to test that hypothesis.

Creating a spatial regime variable

In this section you will divide your study area into two spatial regimes which will be coded as 0 or 1. Thus your spatial regime variable will take on the value of 0 or 1 for each observation.

The method by which you choose the spatial regimes will vary according to your particular dataset. You might want to choose a variable, e.g. income, and divide your study area into a high-income and a low-income regime. Or, if your study area includes counties from two states, you could create a regime variable which identifies which state each county belongs to.

This section outlines a procedure in which you will view an Idrisi map display, and use the mouse cursor to draw a polygon (the selection polygon) on the screen, over your map, such that the selection polygon defines one of your spatial regimes. The selection polygon will be stored in Idrisi as a vector file called SUBSET. You will then run a program that converts the selection polygon file into a 0/1 spatial regime variable, which is input to SpaceStat.

1. If necessary, press **F2** to go to DOS. Use MAPVAL to display a variable of interest, and press **v** to add the vector file of polygon boundaries.

2. Now, without escaping from the screen display, press **d** (for "digitize"). In the lower-left of the screen, you will see a prompt for the feature type "P, L, or A" (point, line, or area). Press **a**, as you will be digitizing a polygon (an areal feature). Next you will be prompted for a feature ID value. Enter **1**. This is the poly-ID value for the selection polygon you are about to create.

3. Now you are ready to use the mouse to define the selection polygon. NOTE: every polygon (e.g. county) on the map that **touches** the selection polygon will have a value of 1 for the spatial regime variable. Map polygons that don't touch the selection polygon will have a value of 0. Therefore when you choose polygons for the "1" regime, you should just touch them with the selection polygon. Don't surround or lasso them, since that would add unwanted polygons to the selected set.

   Using the mouse, move the cursor to where you want the selection polygon to start. Click the **left** mouse button to start the selection polygon. Continue to move the cursor and click **left** each time you want to insert a vertex of the selection polygon. When the polygon is defined, click **right** to "close" the polygon and make the cursor go away.

   You will then be asked whether you want to keep this polygon feature. If you do, press **y**. If not, press **n** and go back to step 2.

4. Press **Esc** to clear the screen display, and you will see a prompt to enter a vector file name. This vector file will store the selection polygon you just created. Type **subset** as the vector file name and press **[Enter]**. Press **[Enter]** again at the prompt for a title, as no title is needed.

5. Now that a vector file called SUBSET exists, you can use a program called REGIME to convert it into a spatial regime variable. Type **regime newvar** to run the REGIME program. "newvar" refers to the name you want to use for the new regime variable.

6. If all goes well, the REGIME program will produce two files, NEWVAR.VAL, and NEWVAR.ASC. The file NEWVAR.ASC is an ASCII file which is ready for input to SpaceStat. NEWVAR.VAL is a conventional values file.

   Before going to the next step, type **mapval newvar** to display your new regime variable, and add the vector boundaries (by typing **v**, etc.). If the regime variable looks ok, go on to the next step. If the regimes are not defined the way you want them, repeat steps 1-5.

7. Type **exit** to return to SpaceStat. Use **D-1-1** to convert the ASCII file NEWVAR.ASC to a dataset called TEMP. Next, use **D-3-1** to merge your dataset with the TEMP dataset. Assuming your dataset is called NEVADA, you would type *nevada* for "first dataset", *temp* for "second dataset", and *nevada* for "merged dataset".

Running the spatial ANOVA regression

Now that you've made a regime variable, you can run the spatial ANOVA. This example will do an ANOVA for CRIME in columbus, using the CP regime variable. The CP variable has a value of one in the center region of Columbus and a value of zero for areas in the periphery.

1. In SpaceStat, make sure that the "Indicator variable" and "Idrisi interface" options are set correctly (as in Part One of this lab).

2. Use **R-1-1-2** to go to Regress/Classic/OLS/Interactive.

3. Type the following at the prompts:

Problem file name:	**reg2.prb**	
Data set filename:	**columbus**	
Name for report file:	**[Enter]**	
Matrix file:	**[Enter]**	
Enter option number:	**5**	(ANOVA)
Variable name:	**crime**	(enter dependent variable first)
Variable name:	**cp**	(name of regime variable)
Variable name:	**[Enter]**	
Heteroskedastic variable name:	**[Enter]**	

On the first screen of output, the value for the CONSTANT term refers to the mean of the dependent variable for the "0" regime (the subset coded as zero by the spatial regime variable). The coefficient for the spatial regime variable (which has the suffix "_1") refers to the difference between means for the two regimes. Thus the mean of the dependent variable for the "1" regime is equal to the constant plus the coefficient of the regime variable.

An example will be helpful here. For the ANOVA on crime in Columbus using the CP regime variable, the constant term is 22.93, indicating that the mean for crime in the periphery of Columbus (where CP = 0) is 22.93. The coefficient for CP_1 (where CP = 1), is 24.90, indicating that the mean for crime in Columbus' center is 22.93 + 24.90 = 47.83. Whether the difference between these two means is significant or not is determined by the coefficient of the regime variable. For the Columbus example, the CP_1 variable is highly significant (t-value = 7.8, Prob = 0.00000). Based on this significance, one would infer a strong and significant difference between the mean values for crime in the two subsets of Columbus.

* How does the fit of this model compare to the previous one?

Mapping predicted values

1. Press **F2** to go to DOS.

2. Type **mapval crime_yp** to map the predicted values from the ANOVA model. Press "**c**" and examine the values in each regime. These predicted values are equal to the mean crime value for each spatial regime.

PART THREE - TREND SURFACE REGRESSION

Trend surface regression is a special type of regression modelling which is best suited for simple smoothing, filtering, or interpolation of data. Trend surface regression predicts each observation of the dependent variable based on the observation's position in space, or more specifically, based on the observation's Euclidean coordinates (X,Y)[1]. The predicted values of such a regression represent a general trend in the data, known as a *trend surface*. Residuals from the trend surface are local deviations from this general trend. The explanatory variables of a trend surface model are terms of a polynomial in X and Y. The *order* of this polynomial reflects the complexity of the resulting trend surface.

For this section, the formula notation will be slightly different. The dependent variable will be denoted by **Z**, and **X** and **Y** will represent variables of the X and Y coordinates of each observation. **a**, **b**, **c**, **d**, **f**, and **g** denote parameters to be estimated by the regression, and **e** denotes the regression residuals. Using this notation, a *first-order trend surface* has the equation:

$$\mathbf{Z = a + bX + cY + e}$$

This equation fits a plane through the dependent variable. An example of a planar trend surface might be the average annual temperature over California, which generally increases with decreasing latitude. This temperature surface could be modeled by a plane which slopes downward from North to South across the state. Residuals from the plane would represent local aberrations from the general pattern of temperature, such as mountains and deserts.

A *second-order trend surface* utilizes a quadratic polynomial equation, and models a simple concave or convex surface. The second order equation is:

$$\mathbf{Z = a + bX + cY + dXY + fX^2 + gY^2 + e}$$

An appropriate dependent variable for a second order model might be the Columbus CRIME variable, which is generally high in the central areas and low in the outskirts. Thus the predicted trend surface would probably be a convex surface with its high point centered over the central part of Columbus.

In this section, you will estimate a second-order trend surface for crime in Columbus, and map the residuals and predicted values.

1. Type **exit** to return to SpaceStat.
2. Use **R-1-1-2** to go to Regress/Classic/OLS/Interactive.
3. Press **F1** to set the following options:

   Option 4: Indicator variable (**ID**)
   Option 5: Idrisi interface (**YES**)

4. Type the following at the prompts:

---

[1] For polygon data, the centroid will be used to represent each polygon's location.

Problem file name:	**reg3.prb**	
Data set filename:	**columbus**	
Name for report file:	**[Enter]**	
Matrix file:	**[Enter]**	
Enter option number:	**2**	(Trend surface)
Constant term:	**[Enter]**	
Variable name:	**crime**	(enter dependent variable first)
Variable name:	**x**	
Variable name:	**y**	
Variable name:	**[Enter]**	
Order of trend surface polynomial:	**2**	
Heteroskedastic variable name:	**[Enter]**	

* Compare the fit of the trend surface model to that of the previous models.

* Map the predicted values using the procedure you used before. Do you see a smoothed pattern?

* After you map the predicted values by typing "mapval crime_yp", use the Idrisi ORTHO command on the TEMP1 image to make a 3-d perspective view of the predicted values. Can you see the convex surface?

* Explore the residuals (CRIME_E.VAL) using both the MAPVAL and RESIDS functions. These are the deviations of each polygon from the general trend in the predicted values.

Further exploration

* Compute a first-order trend surface for crime. How does the model fit compare with the second-order surface? Can you see the planar surface when you map the regression output?

* Map the crime values with MAPVAL. Now run Idrisi's trend surface module, TREND, on the TEMP1 image. Display the output image with the COLA command. What is different about this surface? How well does the trend surface appear to be extrapolating crime values in areas far from the Columbus polygons? (i.e. outside the study area)

SETTING UP YOUR OWN DATA IN THE D:\TEMP DIRECTORY

When you work on your own data, you will need certain files in your work directory D:\TEMP. After making sure that the computer is configured for SpaceStat, type:

```
172load   mydata
```

This will copy the files needed for running SpaceStat and Idrisi. Next, you need to copy your data from your diskette to the D:\TEMP directory.

Make sure you copy, from your diskette, your SpaceStat dataset (.DAT and .DHT files), your POLYID image (both of its parts, .IMG and .DOC), your vector boundary files (.VEC and .DVC), and any other important files (values files, dummy images, etc.).

FINISHING A LAB SESSION ON YOUR OWN DATA

When you are ready to quit the lab session, remember to copy newly-created files back to your diskette.

APPENDIX 1: Installing data from the included diskette

This Appendix shows you how to install the programs and data from the included diskette to your computer lab. The basic procedure is to copy the data and program directories to directories on your network server drive (or a local hard drive if not networked) and to modify a few batch commands so that they contain directory path information corresponding to your system. The following steps guide you through this procedure.

CONVENTIONS: This procedure assumes you are installing data onto a fileserver in the directory E:\LAB\STATDATA, and that executables are to go in the directory E:\LAB\STATPROG. Substitute your chosen directory names below.

1. REQUIRED SOFTWARE: You must have Idrisi version 4.0 or higher and SpaceStat version 1.0 or higher installed in your lab.

2. COPY FILES FROM THE DISKETTE: Place the diskette into the floppy drive (assumed to be A:) and type:

   ```
   a:\> copy  172bin\*.*  e:\lab\statprog
   a:\> xcopy  172setup\*.*  e:\lab\statdata  /s
   ```

   This will copy a set of executables into E:\LAB\STATPROG and seven directories of data into E:\LAB\STATDATA.

3. UPDATE TWO BATCH FILES WITH YOUR DIRECTORY PATH INFORMATION: Use a text editor (e.g. the Idrisi EDIT module) to modify the following two files in the E:\LAB\STATPROG directory:

   a) Edit the file **172LOAD.BAT**. The information you need to update is indicated in bold text on the lines from the 172LOAD.BAT file below:

   ```
   set dpath=e:\lab\statdata\
   set path=e:\lab\statprog;e:\idrisi;c:\dos
   ```

   The "set dpath" line should reference the directory which contains the subdirectories LAB1, LAB2, etc. The "set path" line should include the directories containing your \172BIN\ executables, Idrisi modules, and DOS commands. Additional items may be added to this path, if desired. Rebooting the computer will set your path back to its original configuration.

   b) Edit the file **SPACE.BAT**, so that it references the directory on which you have installed SpaceStat (C:\SPACE is assumed). Information you need to change is indicated in bold text on the line from the SPACE.BAT file below:

   ```
   c:\space\gsruni c:\space\space.gcg
   ```

4. CONFIGURE YOUR SYSTEM SO IT CAN RUN BOTH SPACESTAT AND IDRISI. For Idrisi, the only requirement is that a mouse is installed. For SpaceStat, you should maximize on the available memory and make sure that the ANSI.SYS device is installed. See the SpaceStat manual for more detailed information.

APPENDIX 2: Documentation of linkage commands

This section describes the syntax and function of all of the linkage commands which are used in the labs. All commands are executed at the DOS prompt by typing the name of the command, and all take one argument. Below, command arguments are enclosed in angle brackets "<...>". For all commands which take the name of an image file or a values file as an argument, the .IMG or .VAL filename extension should not be typed. Some commands require certain files to be present in the current directory, for example the POLYID image which is composed of the files POLYID.IMG and POLYID.DOC.

NOTE: The following commands make use of the executable C-programs included on the diskette. Make sure that these .EXE executables are on the current DOS search path.

172LOAD

Usage: 172load <lab1, lab2, ... lab7, mydata>

Description: 172LOAD first clears out the D:\TEMP work directory. Then it installs a dataset for one of the labs 1-7. If students are working with their own data, they must run 172LOAD with the "mydata" argument in order to copy the required setup files to the D:\TEMP directory.

COL / COLA

Usage: col <Idrisi_image> cola <Idrisi_image>

Description: These commands are shortcuts to the Idrisi commands COLOR and COLOR A, respectively.

Required files: BLURED.PAL

DUMMY

Usage: dummy <val_file>

Description: DUMMY converts a .VAL file into a "dummy image", which is an Idrisi image containing one grid cell per observation in the .VAL file. A dummy image is not appropriate for map display, but is necessary in order to get a correct histogram with the Idrisi HISTO command. The output dummy image has the same name as the .VAL file, but with an "x" at the beginning. If necessary, the new image filename will be truncated to eight characters. Future versions of Idrisi will probably allow HISTO to work directly on .VAL files and therefore obviate the need for the DUMMY command.

NOTE: If the Idrisi REGRESS command (using the "values files" option) gives an "Insufficient memory for this module" error, making dummy images out of the input variables and running REGRESS on the dummy images might work around the memory problem.

GZ

Usage: gz <val_file>

Description*:* The GZ command requires a .VAL file containing Z-values for the Getis-Ord Gi statistic. GZ shades polygons with negative Z-values blue if they are significant at p = 0.001, 0.01, or 0.05. Polygons with positive Z-values are shaded red if significant at the same levels. All other polygons are shaded gray. The output image is called TEMP1.

Required files: GZ.PAL, GZ.RCL, POLYID.IMG, POLYID.DOC

MAPVAL

Usage: mapval <val_file>

Description: MAPVAL inputs a .VAL file containing some variable, and produces a color, autoscaled Idrisi map display of the variable, including a legend of the choropleth class breaks. The output image is called TEMP1.

Required files: POLYID.IMG, POLYID.DOC, BLURED.PAL

REGIME

Usage: regime <output_val_file>

Description: REGIME assumes that you have used Idrisi's on-screen digitizing function to create a vector file (area feature type) called SUBSET, which defines a spatial subset of an image (coded as 1). The REGIME command produces a spatial regime variable in two formats: one is a .VAL file which can be displayed with the MAPVAL command. The other is a .ASC file which is in a format readable by SpaceStat's "ASCII to dataset" function (D-1-1).

Required files: SUBSET.VEC, SUBSET.DVC, POLYID.IMG, POLYID.DOC

RESIDS

Usage: resids <val_file>

Description: RESIDS inputs a .VAL file of regression residuals and produces a choropleth map display. The .VAL file can be displayed directly with the MAPVAL command, however the negative values cause the display to look strange. The RESIDS command reclassifies the residuals to positive values, producing a map which is often easier to interpret, although the actual residual values are lost. The output image is called TEMP1.

Required files: RESID.PAL, POLYID.IMG, POLYID.DOC.

VAL2SPC

Usage: val2spc <in_val_file> <out_asc_file>

Description: VAL2SPC converts a .VAL file (created by Idrisi's EXTRACT module) into a .ASC file which is readable by SpaceStat's "ASCII to dataset" function

(D-1-1). This command is useful when Idrisi is used to create a new variable which then needs to be input to SpaceStat for analysis.

APPENDIX 3: Custom Idrisi color palettes, etc.

BLURED.PAL

```
 0  0  0  0
 1  0  0 63
 2 30 30 63
 3 40 40 63
 4 50 50 60
 5 40 40 50
 6 30 30 40
 7 20 20 25
 8 20 20 20
 9 25 20 20
10 40 30 30
11 50 40 40
12 60 50 50
13 63 40 40
14 63 30 30
15 63  0  0
```

GZ.PAL

```
 0  0  0  0
 1  0  0 63
 2  0  0 43
 3  0  0 23
 4 50 35 35
 5 63 30 30
 6 63  5  5
 7 32 32 32
 8  0  0  0
 9  0  0  0
10  0  0  0
11  0  0  0
12  0  0  0
13  0  0  0
14  0  0  0
15 63 63 63
```

RESID.PAL

```
 0  0 10  0
 1  0  0 63
 2 30 30 63
 3 40 40 63
 4 50 50 60
 5 40 40 50
 6 30 30 40
 7 20 20 25
 8 25 20 20
 9 40 30 30
10 50 40 40
11 60 50 50
12 63 40 40
13 63 30 30
14 63  0  0
15 63 63 63
```

GZ.RCL

```
1 -1000 -3.09
2 -3.09 -2.33
3 -2.33 -1.65
7 -1.65 0
0 0 0
7 0.00001 1.65
4 1.65 2.33
5 2.33 3.09
6 3.09 1000
-9999
```

HCPREP.RCL

```
15 0 0
14 1 4
10 4 7
3 7 10
4 10 13
1 13 16
-9999
```

NOTE: .RCL files are used when the Idrisi RECLASS command is run in batch mode.

APPENDIX 4: Converting scanned maps to clean Idrisi raster polygons

Scanning allows virtually any hard-copy map to be imported into Idrisi. One source of maps suitable for scanning is to trace the desired boundary lines from an atlas onto a blank sheet of paper. This section outlines a procedure for scanning a map of polygons and converting the scanned image into a clean set of raster polygons in Idrisi format. Here "clean" means that no border pixels exist between polygons.

NOTE: The map which is scanned should contain nothing but polygon boundary lines[1]. It should be scanned as a bitmap (black/white), and not as grey tones. (Capitalized words below are IDRISI commands.)

1. Scan the image at resolution high enough so that the boundary lines are at least a couple of pixels wide (so that each polygon is separated from other polygons by border pixels. Convert the image to idrisi format. Idrisi reads "tif" format (TIFIDRIS).

2. Use GROUP to assign a value to all groups of contiguous pixels.

3. Use CONTRACT to shrink the image (by pixel thinning) to the desired resolution. This should also remove most spurious groups which may have been created from scanned 'dirty spots', etc.

3a. If a lot of very small spurious groups remain, you can use FILTER with the 'mode filter' option to remove many (hopefully all) of these.

    Now the image is mostly correct, except there are border cells between polygons that we want to remove. This will be done by using ALLOCATE to allocate these cells to the polygon nearest to them. A few pre-processing steps are necessary before running ALLOCATE:

4. Use RECLASS to reclassify border cell values to zero, and reclassify background cells to an arbitrarily high value (this is done so that border cells which are closer to the background than to a polygon will be allocated to the background).

5. If the above reclassified image is not in integer/binary format, CONVERT it to this format.

6. Use DISTANCE on the reclassified image. This will assign each zero cell (border cell) a value equal to the distance to the nearest non-zero cell. This image is needed as an input to the ALLOCATE module.

7. Use ALLOCATE to allocate border cells to polygons. For input, use the image from step 6 as the 'distance image', and use the image from step 4 (or 5) as the 'target image'.

8. Now use RECLASS to reclassify the background pixels to zero (any other desired value).

---

[1] Use whiteout to remove unwanted features before scanning, or use a traced set of map boundaries.

The image should now contain raster polygons with no border pixels between them. If desired, you can now run GROUP again to renumber your polygons continuously from 1-n. Or, use RECLASS to reclassify polygon values to your desired polygon-ID values.

APPENDIX 5: Producing hard-copy Postscript output from Idrisi

This section outlines a procedure for printing hard-copy Idrisi maps and graphics on a Postscript printer. The procedure requires the programs CorelDraw and MicroSoft Paintbrush, and is very specific to UCSB Geography's PS/2 lab. It is included here as an example of capturing Idrisi screen output. If you use this procedure, you will need to modify the files MSPAINT.BAT and CAPTURE.BAT (in your directory of executables) so that they contain the correct directory path to your MSPaintBrush program.

How to make hard-copy maps from IDRISI

INTRODUCTION

This handout outlines the procedure needed to produce printed maps and histograms from Idrisi in the Geography Department's PS/2 lab (Ellison 2612). Since only Postscript printing is available in the PS/2 lab, and Idrisi doesn't support Postscript printing, you will have to bear with the tedious and unintuitive set of steps outlined in this document.

What you can and can't print

You can print simple maps, like the ones displayed by MAPVAL, and histograms made with the HISTO command. All output is black and white--maps will be converted to five shades of grey when printed. Due to memory limitations in the lab, you can not print scatterplots. At the print pre-processing stage, you will have an opportunity to add high-quality text and additional graphics such as scale bars to your maps.

Overview of the hard-copy procedure

The next section contains an in-depth description of the printing procedure. Here is a brief overview of the steps you will follow:

- Run a function called **CAPTURE** to place a screen-capture program in memory.
- Display the desired graphic in Idrisi, using **HCPREP** for maps or **HISTO** for histograms.
- Press **Shift+PrintScreen** to run a screen-capture program and save the graphic display as a **.PCX** file.
- Start up a graphics program called **MSPAINT**, and load your .PCX file. Use MSPAINT's paint roller tool to change the background color of your map or histogram from black to white. Use MSPAINT's eraser tool to erase text, legend boxes, etc.. Save the modified .PCX file and quit.
- Start **CORELDRW**, a graphics arts package, and import the .PCX file. Use CORELDRW to write and position high-quality text for your map titles and annotation.
- Print your finished map from within CORELDRW.

DETAILED PROCEDURE FOR MAKING HARD COPY

Capturing the screen display

1. At the D:\TEMP> prompt, type **capture** to load the screen-capture program in memory.

2. For histograms, use **HISTO** to display the desired graphic. For maps, use the **HCPREP** function to prepare a values file and display the prepared image. The HCPREP function expects a values file name, so you would type:

   hcprep <values_file> (don't type the .VAL extension)

   IMPORTANT: For histograms, write down your minimum value and class width on a sheet of paper. You will be erasing the Idrisi labeling on the X-axis and inserting much better quality text later on. With the minimum value and the class width, you can compute the class breaks of your histogram.

3. Now you should have either a map or a histogram on the screen. If it is a map, press **v** and add your vector boundaries, using color 8. Do not clear the screen display yet.

4. Press **Shift+PrintScreen**, and you should see a small menu in the upper-left of the screen. Use the arrow keys to highlight the **SAVE** option and press **[Enter]**. At the prompt, type **hardcopy.pcx**, or any other desired filename, so long as it has a **.PCX** extension.

Using the Microsoft Paintbrush program

5. Wait ten seconds to allow the screen capture program to save your HARDCOPY.PCX file. Then press **Esc** and type **mspaint** to start the Microsoft Paintbrush program.

6. Click on the **Page** menu and select the **Load from:** option. Click on the line directly under the ">>" symbol, and type **d:\temp\hardcopy.pcx**. Click on **OK** and you should then see your graphic on the screen. Don't worry about the seemingly random colors, these will be automatically translated to a logical grey scale.

7. Click on the paint roller icon, and select the color white from the color/pattern boxes on the bottom margin of the screen.

8. Click on the **Page** menu and select the **Show screen** option, so you can see your entire graphic. Now click anywhere on the black background, and it should change from black to white. For maps, also click on all parts of the yellow background (not yellow polygons) so that the entire yellow background turns black. When the background looks right, press **Esc** to re-display the menu.

   If the color is wrong or if you painted the wrong thing white, press **Esc** and go back to step 6.

9. For maps, skip to step 10. For histograms, you need to erase text from your graphic, so do this step. Click on the eraser icon (just above the paintbrush icon). Click on the **Page** menu and select the **Show screen** option, so you can see your entire graphic.

Use the eraser to erase any all text, including the title and the labelling on the axes. Press **Esc** when done, so that you see the menu.

10. Your graphic should now have a white background, and no remnants of text or labelling. Click on the **Page** menu and select the **Save as:** option. Click on **OK** and then **YES** to write over the existing .PCX file. Now click on **Page/Quit** to exit the MSPAINT program.

Using the CorelDraw program

11. Type **coreldrw** to start the program (note that it is "coreldrw", not "coreldraw"). You will have to reboot the machine with **Ctrl+Alt+Delete**. After rebooting, type **cd temp** and type **coreldrw** again.

12. In CorelDraw, click on the **File** menu and select **Import**. Double-click on the **.PCX** import option. In the *File* field of the current window, enter the full path and name of your .PCX file (e.g. D:\TEMP\HARDCOPY.PCX). CorelDraw will take some time to import the file. Once imported, your image will be visible, but you cannot modify it. That's why you used MSPAINT to pre-process the image.

13. To get a better screen display, click on the **Display** menu and select **Show Preview**. This will split the display into two screens. The left side is where you point and click to add text to your map, and the right side is where you see a WYSIWYG[1] display. (Again, the colors you see will be printed as logical grey tones.)

14. Now click on the text icon from the left margin (the icon is a letter "A"), and click on the place you want your map or histogram title to go. You will then see a window in which you enter your text and select text options such as font and point size. Enter your text and click the **OK** box. Your text should appear on the screen display.

    Once on the map, text can be selected and moved around with the selection tool (top icon on the left margin--looks like an arrow). You can also edit existing text by selecting it and choosing **Edit text** from the **Edit** menu.

    CorelDraw has a thousand other functions that obviously can't be listed here. Ask someone who knows CorelDraw for more information.

15. Place all the text and annotation that you want for your map or histogram. For the histogram, be sure to label the X-axis with the class breaks. For a map, you might want to add statistics such as minimum/maximum values, mean, standard deviation, etc. You should definitely include annotation which describes the logic of the grey tones on your map, e.g. "darker tones represent higher values".

16. When you are ready to print, click on the **File** menu and select **Print**. Click on **OK** and your graphic will be sent to the printer. The printing process takes up to 15 minutes. Be patient. If nothing has been printed for 20 minutes, you might have to try running CorelDraw on another machine and printing from there. In any case, save your graphic by clicking the **File** menu and selecting the **Save as** option. Enter a filename of your choosing with a .CDR extension (e.g. hardcopy.cdr). Make sure to save your .CDR files to diskette, in case you need to modify or re-print them.

---

[1] "What You See Is What You Get".

APPENDIX 6: Extracting student datasets from the ARCUSA CD-ROM

The following procedure assumes that the ARCUSA CD-ROM is mounted on a Unix directory called **/cdrom**. In addition, it is assumed that new datasets are to be written to the workspace **/a/toast/u1/dodson/arc/arcusa**. If these paths are not correct, the macros GETSUBCOV and PULLJOIN will have to be modified. The lines needing modification are clearly marked in the source code with the following line:

```
/***** INSERT YOUR DIRECTORY PATH HERE *****
```

EXTRACTING A SUBCOVERAGE FROM THE ARCUSA CD-ROM TO AN IDRISI IMAGE

Extracting Spatial Data

1) In Arcplot, extract a small subcoverage from one of the large CD-ROM coverages:

arcplot: GETSUBCOV <incov> <outcov>

- <incov> is either an existing Arc coverage in the CD-ROM (slow) or a copy of a CD-ROM coverage in the user's workspace directory (faster).
- user is prompted to lasso polygons of interest
- this can be done from the cd workspace
- writes selected polys to ~dodson/arc/arcusa/<outcov>

2) In Arc, convert the subcoverage to an Idrisi-readable format (Idrisi reads Erdas files):

arc: COVIDRISI <incov> <out_erdas_file> <cell_size>

- <incov>: subset coverage made with GETSUBCOV
- <out_erdas_file>: filename for erdas file (to be read by Idrisi)
- <cell_size>: in coverage units (typically meters)
- COVIDRISI will report the number of rows/columns in the erdas file. If the row/col numbers are not satisfactory, enter 'q' to quit and run COVIDRISI again with a different <cell_size>.
- The output erdas file contains polygon-id's equal to the <incov># attribute from the <incov>.pat file.

3) In Unix, move the binary Erdas file to a DOS diskette: (The following is for the AIX operating system. Your operating system may use a different procedure.)

% doswrite <erdas_file> <dos_erdas_file>

- writes <erdas_file> to a DOS diskette in binary format

4) In Idrisi, Import the Erdas file:

idrisi: ERDIDRIS

- Idrisi will prompt for needed information
- Use the Idrisi DOCUMENT module to check the parameters of the new Idrisi image. Change any strange values (e.g. strings of garbage characters).

5) In Idrisi, set background cell values to zero:

idrisi: RECLASS

- Reclassify all old values of -9999 to a new value of 0 (zero).

Extracting Attribute Data

1) Find out what cdrom coverage contains the needed attributes. (e.g., POP90C, POP88C, GOV88C, SOC88C, ENVIR, AGIN_C, AGVL_C. These are the 1:2M county-level coverages.)

2) Pull items (at most seven at a time) from the cdrom coverage and join them to the subset coverage:

```
arc:  PULLJOIN <cdrom_cov> <subset_cov> <num_items> <item1> <item2>
      . . .
```

- <num_items>: # of items to follow (maximum of 7)
- <item1> ... etc.: up to 7 item names from <cdrom_cov>

3) Repeat steps 1 & 2 above until all needed attributes are in <subset_cov>.pat.

4) Now make an ASCII file of the attributes:

```
arc:  GETDATA <subset_cov> <outfile> <num_items> <index_item>
      <item2>  ...
```

- <outfile>: name of ASCII file of attributes
- <num_items>: # of items to follow (maximum of 10)
- <index_item>: should be <subset_cov># (very important)
- <item2> ... etc.: up to 9 item names from <subset_cov>
- NOTE: The <index_item> is the attribute which links the polygons to their attributes in IDRISI and SpaceStat.

5) Using a text editor, add header information to the top of the ASCII dataset file. The header format is as follows:

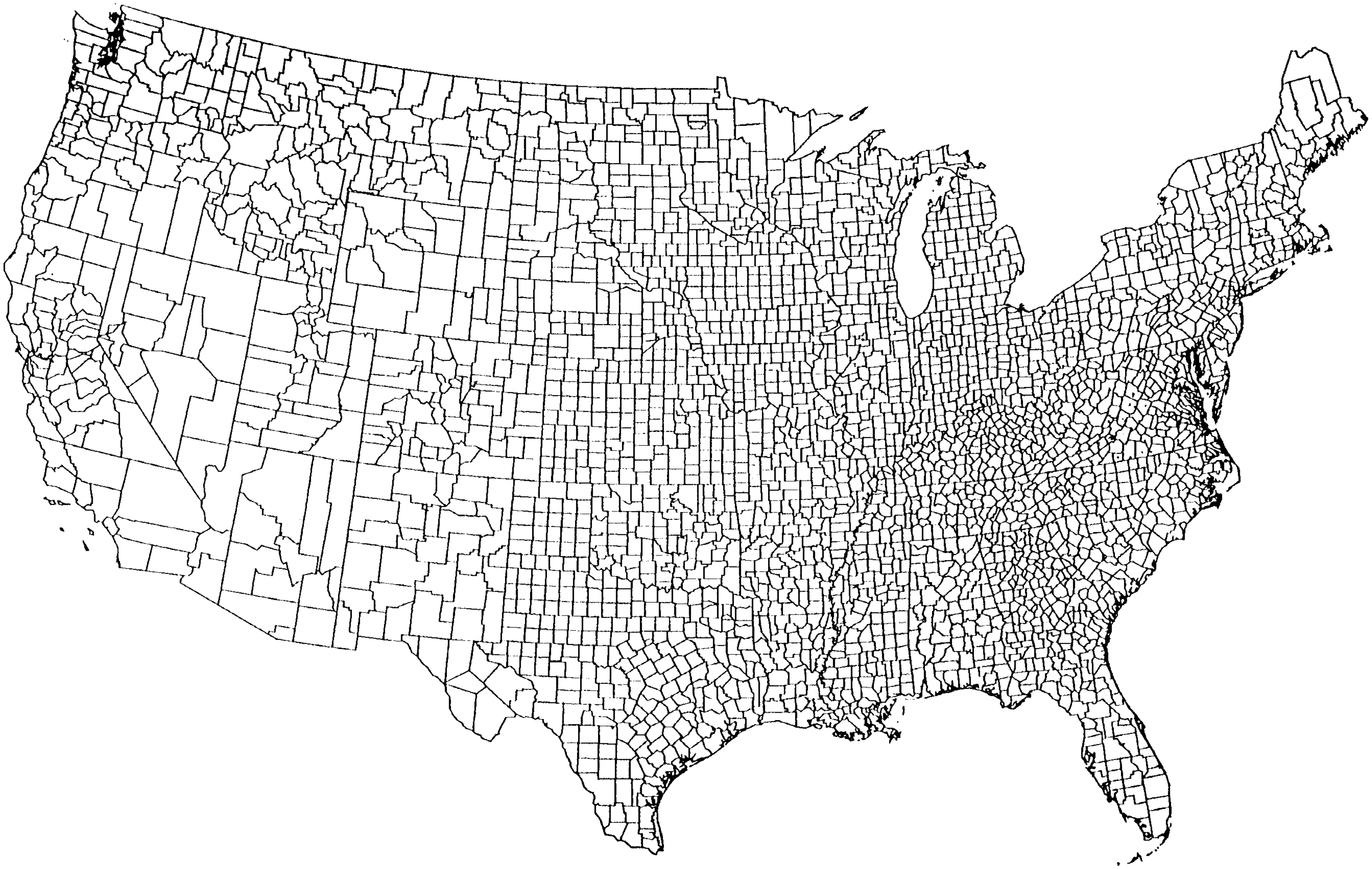
```
<num_obs> <num_vars> <var1_name> <var2_name> ... <varK_name>
```

- Above, <num_obs> is the number of observations (N) in the dataset; <num_vars> is the number of variables (K); the remaining K items are the names for each variable.
- The header may take up more than one line of the file, if necessary.
- Variable names should be 8 or fewer characters in length.
- Your polygon-ID variable (<index_item> from step 4 above) should have the name "ID".

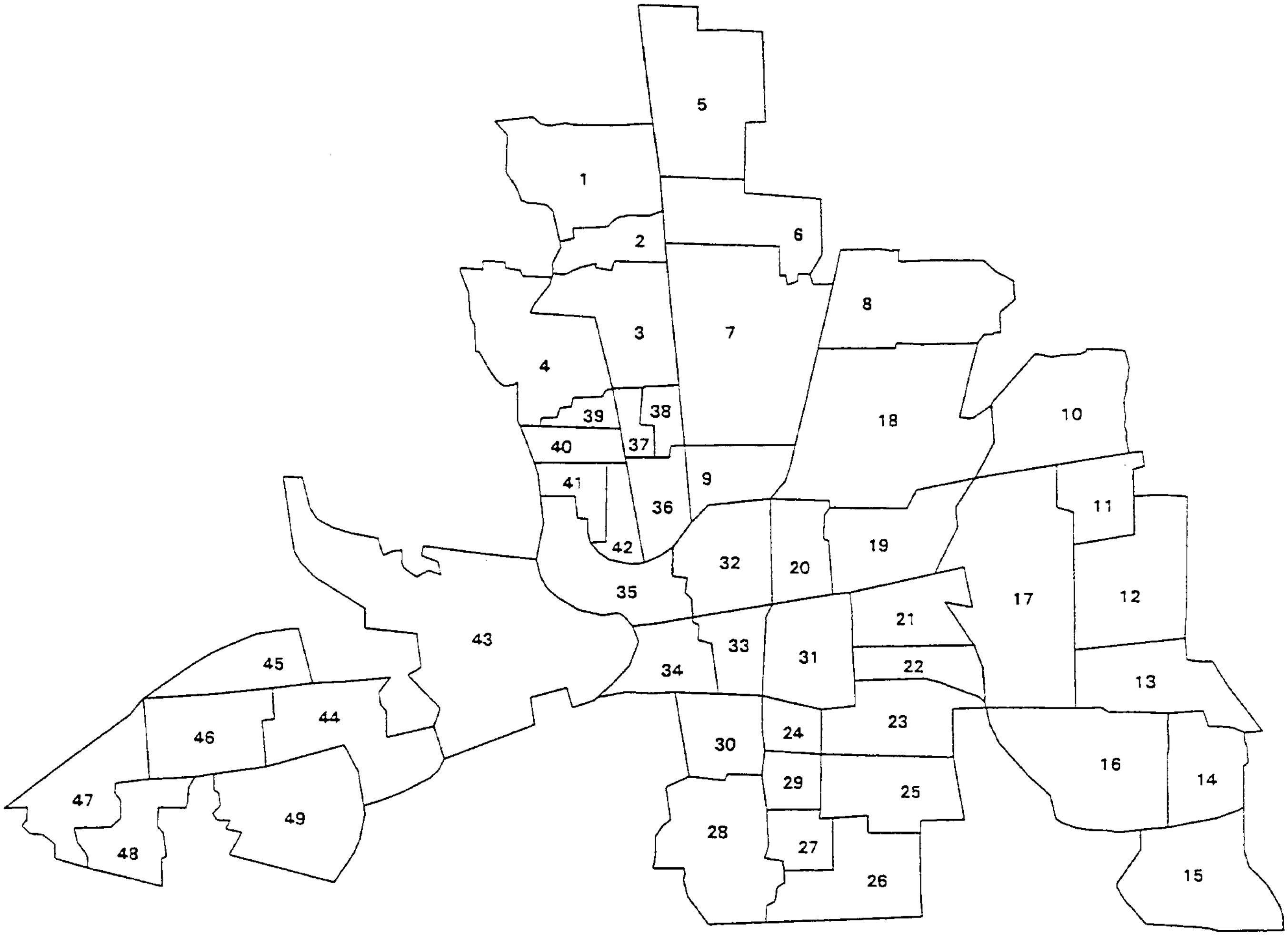
6) To copy the ASCII dataset file to DOS (using AIX), type:

```
% doswrite -a <unix_file> <dos_file>
```

You should now have an Idrisi image of polygons and a corresponding ASCII dataset. These files are required as the point of departure for Lab 2 (Introduction to SpaceStat).



The 3,111 counties of the 48 contiguous United States



Columbus, Ohio neighborhoods