Geek Like Me

The layman's guide to Computer-Assisted Reporting

By Matthew Waite

Table of Contents

NOTE: This book was written using a Windows 95 machine. The software used was Excel 97 for spreadsheet exercises, Visual FoxPro 5.0 for databases and Maptitude 3.0 for mapping. I am not an employee of Microsoft Corp. or Caliper Corp. and am not endorsing their products over any others. These were the programs that I settled on after gathering recommendations from other CAR savvy reporters. While specific instructions for other spreadsheet and database programs will be different than those in this book, the general techniques held within are good for all platforms.

Welcome to Geekhood: A Brief Introduction

In a cavernous hotel ballroom, with a cheap folding table serving as a mic stand for a panel of speakers discussing the interplay between journalists and public officials who hold electronic records, a geek joke was told.

It was a fitting room for it. It was the national convention of the National Institute of Computer-Assisted Reporting. It was the last night of the convention, and the room was packed.

Laura Matthews, who was a shade young for a projects reporter, but gifted with computer-assisted reporting skills at the Nashville Tennessean, was explaining how she had requested a database from another in a series of government agencies as a member of the panel. Matthews was met with a scoff. "Oh no. You can't have that. It's in a really complicated program."

Matthews persisted, asking what program the database was stored in. The bureaucrat told her that it was so complicated that she would never be able to get the database out. She still persisted, wanting to know the program.

"And he said 'Quattro Pro,'" Matthews said.

The room erupted in laughter.

Just then (I don't remember his name: I am working on it), who was moderating the discussion, came to the podium.

"It could only be this group that the punchline to a joke could be Quattro Pro," he said. Again, the room erupted in laughter.

Miss something? Wondering how a group of adults could laugh so heartily at the mere mention of a commercial database program that you can pick up at any low-rent computer store? Wondering who could find such a thing funny?

Geeks.

Proud ones.

And if you are reading this, you want to be one – or some semblance thereof.

In today's newsroom, the word "geek" is no longer a putdown. It is a badge, one of honor and sacrifice. Geeks do the "geek work" on projects. They are the ones generating charts and graphs and spreadsheets for the graphics department to make pretty. They are the ones doing the number crunching, the database massaging, the spreadsheet labor.

It isn't sexy, and it takes a major discovery to get the adrenaline pumping, so it takes a patient breed to get into geek work. But computer-assisted reporting, or CAR, has become a phenomenon in newsrooms across the country that editors are clamoring for. They want numbers, they want depth, they don't want to get scooped by the other guy. And they are increasingly looking for people with the skills to pull off CAR work.

Quite simply, computer-assisted reporting is using computers teamed with spreadsheet, database, statistical, mapping and Internet software

Newsrooms all across the country are getting more and more involved in CAR. Management is starting to put money into CAR at many newspapers and others are starting to hire database editors to spearhead CAR projects.

Newspapers like the Cincinnati Enquirer and the Arkansas Democrat-Gazette have hired full-time database or computer-assisted reporting editors within the last year and a half. Other newspapers, such as the Wilmington Morning Star in Wilmington, N.C., have assigned city editors to keeping an eye on CAR. In other words, it isn't just the big boys playing with the big toys. Smaller and medium sized papers are getting involved with CAR. Broadcast outlets are even getting in on the action. Several stations with investigative reporting traditions have started using databases and spreadsheets to generate good television.

Diane Graham, the managing editor at the Des Moines Register, set up a committee of newsroom personnel to establish what that paper needed to get into the CAR game. The committee came up with a dozen points, with ideas ranging from hiring a database editor to staff training.

But it doesn't have to be so hard.

Computers are ugly, nasty, unyielding, complicated creatures that can frustrate you to your very core. They are also powerful tools, even in the hands of the greenest neophyte.

Geeks are the folks in your newsroom who dig numbers, sort of. You would be hard-pressed to find a math major in your newsroom. Geeks are the folks who have screen loads of numbers up when people walk by, shaking their heads.

But CAR is not voodoo. It never has been. In the early 1980s, a few forward thinking journalists figured out that they could take computer databases of bank lending data and ask it questions with a computer. What they produced was good enough to win a Pulitzer Prize. "The Color of Money" ran in the Atlanta Journal-Constitution. It was a

story about Home Mortgage Disclosure Act data, and they found that blacks were denied bank loans at a far higher rate than whites.

And CAR was born.

The craft has remained in the hands of a chosen few – and that small group has shown steady, but small growth – but that is changing. On any given week, one can open up Editor and Publisher and read the series of reporter openings that say "computer assisted reporting experience a plus." CAR skills are in demand. And it takes surprisingly little to get in on it.

Don't believe me?

Well allow me to show you a thing or two. You won't be programming in SQL server setting up newsroom intranets through a web browser or stripping out extraneous code by using FoxPro string functions by the end of this, and there is a simple reason for that. I can't do it either. I am no expert. I am a student. A senior. Not far from graduation. But in my schooling, I decided I wanted to learn CAR, and so I did. On my own mostly, but others had a hand in it. I sat down with books, websites, databases, deadlines and desire. And what follows is what I learned.

But the beauty of CAR is that even though I don't have a title like "database editor" or years of experience or a computer science minor, I have been able to turn my acquired skills into real results.

With CAR, I have proven that a poor neighborhood that everyone thought was a crime-filled ghetto was just a neighborhood of working folks and students, and had one of the lowest crime rates in the city. With CAR, I have halted city government studies on urban renewal so they could rethink things. With CAR, I have shown a bank that there

was serious need for low-income home improvement loans for a neighborhood. With CAR, I have shown students and Lincolnites where crime was occurring on their campus, and in their neighborhoods. With CAR, I have altered police patrolling routes. With CAR, I showed voters who was buying their election through campaign donations. With CAR, I showed Nebraska voters that donors to state campaigns were abusing the same loopholes they were hearing about on the national news. With CAR, while working my cops beat, I showed readers that more than 80 percent of all homicides are domestically related homicides in Lincoln after a boy shot his father and people searched for answers.

So with just myself as a teacher, I have taken a powerful tool and used it just a little bit. Again, I claim to be no expert, but then at this stage of the game, there are very few CAR experts.

I just claim to be a geek. A proud one. My newsroom colleagues walk by my workstation and find statistics textbooks, SQL language manuals, printouts of record layouts, thematic maps and me, staring focused at the language of a query to make sure I got it right. And they just shake their head. "Geek," they say, walking away.

That's fine. I'm proud of it. When my stories run deeper than everyone elses, when the competing newspapers have to run "The Daily Nebraskan reported ..." because they don't know CAR, when I scoop everyone, I am especially proud of it.

And you will be too.

Geek Speak: Learn the Lingo

Probably the most daunting aspect of CAR is the language. If you sit down and talk to a geek for a while, they just start to rattle off acronyms and foreign sounding words. Chances are, you caught them while the were doing geek work. It happens to the best of us.

"Hi Matt. How are you today?" a frazzled editor asks.

"Use TALK; select RESPOND when LANGUAGE = QUESTION." I respond.

"Uh, that's great. You just sit there and wait for the men in white coats, okay?"

What strikes me as truly funny is that the language barrier is the worst of it. After you can understand the lingo and the jargon, it's all downhill.

Really. It is.

I call them the Terms of Endearment. It is just a simple list of common CAR terms that you will hear over and over and over again, and probably just glazed over when they were hurled your way. Well, glaze no more.

They may sound complicated: they aren't.

ASCII

It stands for American Standard Code for Information Interchange and it is pronounced "ass-key." It is what it says, but it is more. It is an unformatted, 256-character set of letters and numbers, but it is the world standard for information exchange. Almost all of the Internet text is ASCII. Any machine can read ASCII. ASCII

is the most versatile language you will use – and oftentimes the most frustrating. We'll get into that later.

Field, cell, record

Very simply, a field and a record are the same thing. A field is generally associated with databases. Cells are commonly associated with spreadsheets. They are one box in a database or a spreadsheet. They are a pool for your data – just one small one in a bigger whole. Think of them in terms of ice-cubes in a tray. Each cube sits in it's own hole, but they make up a larger tray in a pattern.

Delimiter, delimited

A delimiter is a character in ASCII text that signals where that particular field ends and starts. For example, my name is Matthew David Waite. In a database of names that run first, middle, last, and was ASCII text delimited by commas, it would be Matthew,David,Waite. The computer reads that and knows that for every comma, there is a field. There are any number of delimiters – tabs, spaces, semicolons, periods, numerals – but the comma is the most common. Here is your first warning: be very, very wary of anyone offering you databases that are in non-delimited ASCII text. An example: 199601010200381919 N. 27 ST.LINCOLNNE6850332158236548465413216574896 54 Look like Klingon to you? It did to me. That's one record from the Lincoln Fire Department's Fire Incident Reporting System. It was a fire in 1996, January 1, at 2 a.m., in district 38, 1919 N. 27th St. in Lincoln. The string of numbers at the end of the record, from NE on, are internal codes used by the department to record how many and what personnel were used. I had to take a record layout and stick fields in there the best that I could. It took 12 hours.

File Extensions

A working knowledge of file extensions will save you more time that any other tidbit of CAR knowledge when you are sitting down to start the data analysis. File extensions are not something that makes a file bigger. What they are is the three-character tag put on files following a dot. Files like matt.txt, matt.dbf., matt.zip, matt.doc. If you know the extension, and what they mean, that means you know how to deal with the file. The most common file name that you need to know is .txt. Simply enough, it is a text file. Text files are usually ASCII text files, and any computer can read them. Most any file that you save off the World Wide Web, using the save as command, will be a .txt file. We'll get into the usefulness of the web and the handiness of .txt files down the road. Right now, just know that .txt means text.

The next most imporant is going to be .dbf. What that means is a dBase (pronounced dee-base) file, a simple standard database file format. If you receive a file that is a .dbf file, you can just open up your database program and go to work. Spreadsheets read .dbf files also. If you have a file that you are going to want to go back and forth between spreadsheets and databases, a .dbf file is the best way to go, in my opinion. It's never failed me.

Another common one is the .zip file. That just means that the file has been zipped, or compressed. The term "zipped" comes from an old DOS program called PKZIP that is used to compress large files onto floppy disks. Many large files on the Internet will be zipped. Many government agencies will zip the files that they give you to save on the costs of disks. What you will need is a program to unzip the files (Stuffit on the Mac and WinZip on Windows machines are the two that I use. But there are others out there that work just as good.). And then they will go from a .zip file to one without an extension, or

to the file type that you need. Always ask the person giving you the zip file what is going to happen when you unzip the file – such as what will it turn into. If you end up with an ASCII non-delimited text file, you aren't going to be happy. Best to avoid that before it happens.

There are several other extensions out there, all for individual programs, so it is best to know what you are dealing with. File extensions are critical to know when you are dealing with the importing and transfer of files. If you don't know the file extension, then you are going to have trouble. Know the extension – and what it is – and it will be smooth sailing. But we'll get to that later.

Geek Think: Get Inside the Geek Mind

Often, what separates the geek from the regular reporter is merely a change in thought patterns. It isn't complicated knowledge and fancy-dancy computer work, it is just a way of thought.

An example, if you will.

The first day of my internship at the Arkansas Democrat-Gazette was the last day before one of the city editors went on maternity leave. Myself and the other intern were left to wander around the newsroom most of the morning, introducing ourselves to whomever would talk to us. We didn't have a desk, and we already knew the computer system. The three other city editors were scrambling to get that day's paper together and get all their ducks in a row – the interns were the last thing on their mind.

Finally, after lunch, an assistant city editor came over to us and offered us feature stories. Erin, the other intern, got to him first, so she got first pick. She took a story about the booming illegal freon trade nationwide and she was told to localize it. I got another national story on how Americans were increasingly turning to cremation after they died and I was told to localize that. We asked when our deadlines were, thinking they would be the next day. It was Tuesday. He wanted them by Friday afternoon.

We didn't know what to do with ourselves. We were used to the daily grind back in Lincoln. Here we were with four days to kill.

Erin and I set about on our reporting. Erin is a budding geek, but I am full fledged. Our first moves illustrate the geek-think mentality. Erin's first instinct was to get on the phone to mechanics and dealers to get a grasp on the problem. Mine was to get on the phone to the Arkansas Department of Health and find their database of body

dispositions. Then it was to the web for national stats and national sources. Then it was to the phone book for local sources. Later, when I got the state database, I did county by county stats, and then went looking for statewide sources and trends.

In the four days, Erin and I reported the hell out of two localizations. Both of us were pleased with our work, and both of us ended up with our first stories going on 1A, her's on Saturday, mine on Sunday.

The geek think process goes like this: find the people with the data, get the data, work the data, find the people. Got that? You need to find the data – which is usually kept by people, but sometimes you can find it on the web. With the data, you can find the trends and detail that can point you in new directions for your reporting. And then you can find more people. Newspapers are all about people, so CAR without people is just a bunch of unreadable numbers. The last step is the most important. Get the people.

The great thing about CAR is you are only limited by three things: availability of the data, the accuracy of the data and your instincts.

Databases are surprisingly more common than you think. Think of what you did today. Did you see someone getting a speeding ticket this morning? They got entered into a database. Did you get a parking ticket today? That's in a database. The property value of your house? Yep. The demographics of your neighborhood? Yes, it is. The loan you got to fix up your home? Uh, huh. Did you vote this year? Your registration is in a database, as is the fact that you voted, thanks to the Motor Voter bill passed in 1992. Your drivers license information is in a huge database. Get married recently? Your marriage license is in a database. Have a pet? Their license is in a database. Basically,

any time there is paper record, there is a good bet there is a record kept in a database somewhere.

Availability is your first problem. Just because there is paper record doesn't mean there is a computer record. Generally, the smaller the government agency, the less chance there is of the records being computerized. You are more likely to get sentencing databases from large cities than you are from the rural county with 800 people. But like all things, it is a case by case situation. The point is, it never hurts to ask. Ask your beat sources. Ask them what databases they keep. You might be surprised.

Your next surprise will come when you ask for the databases. Sometimes, the government agency will want money for the data. Sometimes they will just deny you the records just out of hand. I had to spend two hours explaining to the University of Nebraska-Lincoln Police Chief, the most decidedly non-geek man I have ever met, that if he gave me a copy of his 1995 incidents database, I would not be able to then break into his computer files and alter reports. It was easy to convince him to give me the data after I explained that without a single modem in the building, I had no way to get access to his computers in the first place. And in some cases, government agents are more than happy to just give you a disk and send you on your merry way. It is all hit or miss.

Using tips that I have gleaned from others who have been down this path and my own personal experience, I have come up with a series of steps that work the best for getting data from the government. But before we start that, you need to know your state's open records act and state open meetings law. That is just basic. If you don't, shame on you, and go learn it.

But if you are trying to get data that you know is there, try these steps.

1. Ask nice. Say please. Tell them who you are, what you want, ask nice, say please and you would be amazed at how often the comply with you. You don't have to get all technical about what you want to do. Just give them the quick and dirty version, and slip in the fact that you are doing this in the public's interest. You would be surprised at how many journalists don't do this. There are many reporters out there who go into offices with official request letters and threats of lawsuits on their hips like gunfighters at the O.K. Corral. When you challenge someone, their first instinct is to rear up in defense of themselves. Chance are, when you ask, they will say no, if you go on the offensive right away. More often than not, you have the right to the data. But you don't need to flaunt that (yet). Another good thing to do is to start with the lowest rung on the data ladder you can get. Start at the data entry supervisor. You would be surprised at how often they just pop a disk in, copy it and hand it over. Conversely, they may send you to another supervisor, who sends you to another supervisor, who sends you to the city attorney, who tells you no, but it is always best to start with the geeks in their department. They are usually so shocked to have someone to talk to that they are incredibly nice.
2. If they do say no to a polite request, then the next step is to write a letter to the supervisor of the data, the head of the department and the department's legal counsel on company letterhead. Get your state's open records act and cite the pertinent sections. Make sure you make the letter clear. Make a specific request and cite specific statutes concerning your request. If it is a federal agency that you are writing to, you need to get a copy of the Freedom of Information Act (FOIA) and send a formal FOIA request to that department's FOIA officer. But make sure you are

specific and clear in your letter. If you leave ambiguity in the letter, they can deny you for requesting too much of the department. In all cases, you should ask for a fee waiver. Under FOIA, you can request a fee waiver as a member of the media. In state government, it changes. Always ask for an itemized bill for any charges to be sent to you BEFORE they do any data work. Decide in advance how much you are going to pay for the database. If you want to, you can include a sentence that says "My organization will pay for data costs up to $50 without approval. If the costs will be greater than $50, we request an itemized bill be sent to us prior to any data processing is done." Many times you will get a disk and a bill for $50. You paid for convenience. The reason you should always request an itemized bill is because some government bureaucrats have decided that data can be a cash crop for their departments. Many charge an incredible amount for their data. The Omaha World-Herald was billed $1.2 million for the state's driver's license data. That equated to $2 a record. Needless to say, the paper didn't bite on the offer. An itemized bill prevents a bureaucrat from just assigning a value to the data. Recently, I requested the Lincoln Fire Department's Fire Incident Reporting System database. I was told that for 1996, it would cost $500. I asked for an itemized bill. In three days, the cost went from $500, the cost the supervisor just threw out, to $100. A half an hour of programming at $50, and a half-hour run time, another $50. By asking for an itemized bill, I saved $400.

3. The final step is the step you don't really want to take. If they deny your written request, you are left with three options. Most states use their attorney general as an arbiter of open records matters. A newspaper can request a ruling from the attorney general regarding the openness of the record. That's option one. The next option is to

sue the government agency in charge of the data. For most small papers and students, this is not an option. Larger papers, which keep the most expensive law firm in town on retainer anyway, this is easier. Sometimes, it takes a judge's ruling to free up a record. Sue for legal costs, as well as the data. The third option is to give up – not what editors want to hear, but it is an option.

That brings us to the topic of charging for data. The plain truth of the matter is most government agencies are going to charge you for something connected to the data. If it's just the cost of the disk, they are still going to charge for it. Every now and again, you will get lucky. But chances are, you are going to get billed for programming time, mainframe, or run, time and the cost of the disk, tape, or whatever they give it to you on. But don't ever just accept the first price as the only price. Argue that these are public records and the public has paid to have them available to the public, and therefore should be free. When that runs out, argue for the cheapest price possible, always reminding your friendly government agent that the data has already been paid for. Tight budgets, they will tell you, won't allow for programmers to just willy-nilly write programs for people. Costs for these vary, and vary widely. It is best to go into the negotiations knowing a few things, like how much a government programmer makes in an hour. It is public record. Find out what they make, so when they tell you that one hour of programming time is going to cost $75, you can ask why that is because the programmer only makes $25 an hour. Then watch as the price falls. Run time costs also vary, but the standard answer I have heard from other journalists has been $40 an hour. One very good way to keep your costs to a minimum is to make your request and the amount of data you want as simple as

possible. The more complicated you get, and the more you want, the more programming/run time you are looking at. For instance, instead of asking for all the property fires for 1990 to present from the fire department, just ask for the whole shooting match and glean out the data yourself. More run time, no programming time. And you never know, in the excess data, there may be other stories to be had.

One last point on data acquisition: get the record layout, get the record layout, get the record layout. Without it, it's like flying a plane with no windows – you are bound to crash. If you don't know what each field is, how are you supposed to analyze the data. Be sure to get the record layout.

And that brings us to accuracy. Accuracy is a scare word for journalists. Accuracy is a strict discipline, an unshakable goal, a hard and fast rule. If it isn't right, don't print it. CAR is no different.

But just because it's in the database, does it mean it's right?

Absolutely not.

If the mayor said that 75 percent of your police force were women, would you just print that without checking it out? Unless you live in a different city that the rest of us, and work at a different kind of newspaper, you would check it out. Chances are, you would find that the mayor omitted a decimal point, or just flat out had it wrong. That is a problem with CAR. Typos can be the difference between you reporting that Sen. John Q. Public raked in $10,000 in one donation, which is not the same as the $1,000 he really took in. A minimum wage clerk at the Federal Election Commission apologizes for the error. That is some of the problem with CAR. Sometimes, the data just isn't put in very carefully. You need to always beware, and always ask the data owner if what you have is

the same results they get when they look at their own data. You need to be always looking for funny looking records. Go check them against the paper copy of the record.

An example of checking accuracy.

For four months, I spent night and day working on a database of the Lincoln Police Department's 1995 incidents file. The problem was the department did not have a standardized method of entering in addresses. The street officers found 14 unique ways to key in Lincoln High School. Common place names were rampant. I hand checked 26,000 records for common place names. Convenience stores were terrible. There are six Kwik Shops in Lincoln. Each one in a different district for the police. So when they had an incident report that listed the address as KWIK SHOP, I had to go down to the records department, ask them for the case number, look up the reporting district and then I would know which one it was. Four months of that. Night and day. Meticulous, nasty work, but well worth it. You will hear about that later, when we get into spreadsheets and databases.

The point is this: just because it is in the computer doesn't mean it's right. Check it. Ask about it. Double check it. Then run it. It's just like any other kind of reporting.

Instinct is the final limitation of the geek mind. It is a bit more difficult to explain, but a better way to put it is news judgement. With a database, you have a world of information on one topic. But, with all that info, you need to separate what is interesting to you, and what is newsworthy.

The best example of this was my first example. At a training conference put on by the National Institute of Computer Assisted Reporting, we were shown a database of hunting accidents in Wisconsin. It was a great database. The reporter who used it first

typed it in himself. It was as clean as databases got. We started to play with it, and we started simple. We found the county with the most hunting accidents. Then we did regions. And then the instructors let us play. By the end of the playtime, I could tell you the brand of shotgun you were most likely to get shot in the buttocks with in Wisconsin's Rock County. Newsworthy? Hardly. Interesting? Yeah, kind of. For no good reason. The point is, just because you have the data doesn't mean you have to bog the story down with more numbers and stuff that is just interesting, not newsworthy.

The World Wide What? Surfing Your Way Into Geekdom

If you have a pulse – I am assuming that you do – then you have heard of, and even used the World Wide Web recently. Well guess what. The web is good for more than checking college football scores on ESPN's Sportzone and keeping anti-pornography groups busy. There is an unbelievable amount of data out there, just for the taking. All you have to know is where to find it, and how to take it off the net.

Also on the Internet are journalism discussion groups – along with about every other kind of discussion group imaginable. In fact, there is a site called Liszt where you can search a database of all discussion groups. The address is http://www.liszt.com/

And yes, there are CAR discussion groups. One of the best is run by the National Institute of Computer Assisted Reporting, or NICAR. To get on it, and if you are going to be doing a lot of CAR work, I strongly recommend that you do, send an e-mail message to listproc@lists.missouri.edu. In the body of the message, type "subscribe NICAR-L (your name)." And voila, you will start getting mail from the list. The list deals with everything from help finding databases, discussions on data costs, to high power programming in structured query language (we'll get to it later) and heavy-duty statistics. It is a great resource for anyone with an e-mail account.

But the web is the primary source for CAR info, including databases to play with. Some sites are set up specifically for downloading databases. American University runs a site where you can download campaign finance information for each district in each state in a .dbf format (remember your file extensions?). It is slick, fast and easy to use.

Included in the packet of data and exercise information is a bookmarks file for your web browser. In that bookmarks file has a bunch of sites that have data, contacts for data and just general information that makes reporting easier.

One thing many reporters don't realize is that using the web is CAR.

Example: I was given an assignment for my magazine writing class on the elderly. It was still a preliminary assignment – the class was going to do a project on the general topic of the elderly in Nebraska. So I got onto a site called Dogpile, which searches all the other search engines for you. I entered in "elderly," "Nebraska" and "statistics" into the search window and in seconds, I had reports, maps, graphics and stats. That's simple CAR.

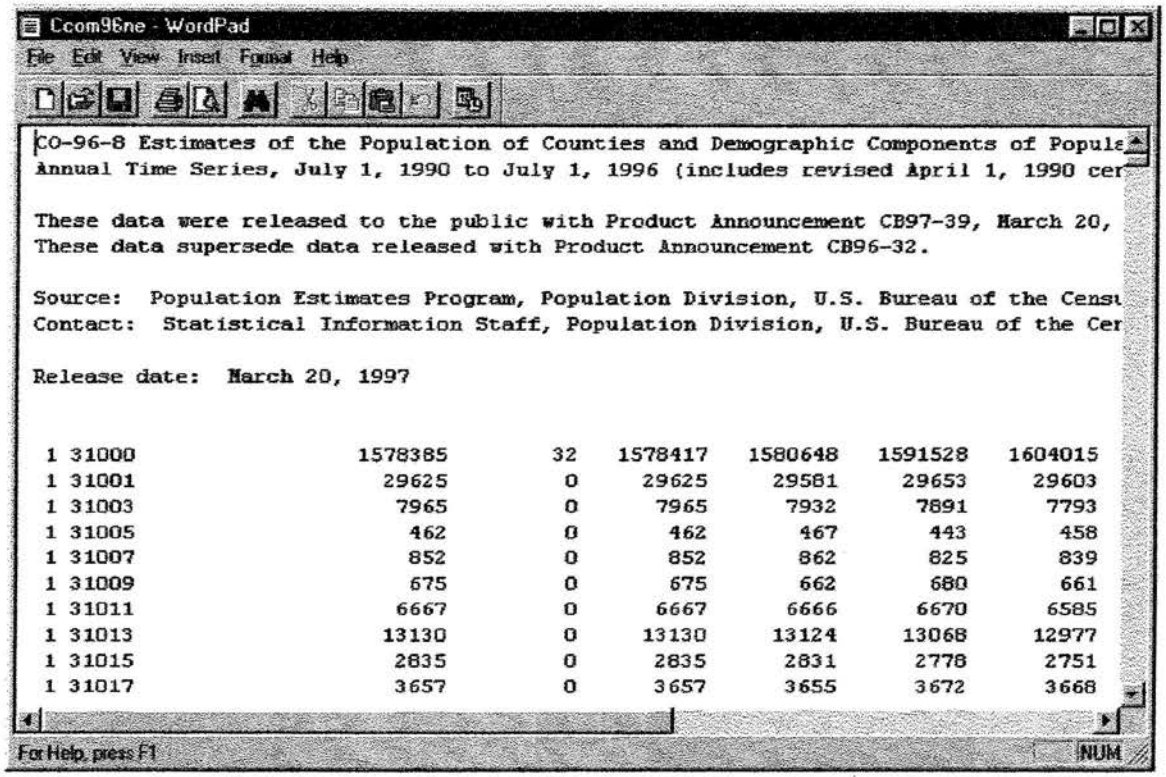
More complicated CAR is taking files off the Internet to use them as databases and spreadsheets.

Here's your first exercise.

Go to the Census Bureau's website. This is a site any journalist should know. It's at http://www.census.gov/

On that site is a button for topics A-J. Let's go to that. Go to P, for populations. Pick the projection link and then go to the county projections. On that screen, go ahead and pick your state off there. I am going to take mine, Nebraska. When you download it, it should come as a text file. If it is zipped, you are going to have to unzip it, and then it will be a text file. While you are here, go back to screen that had the state and county projections link and print out a copy of the record layout. You are going to need that.

Now that you have it on your hard drive, go ahead and open it in your Simple Text or Word Pad programs. It should look like this:



Looks like organized gibberish, doesn't it? That's why you downloaded the record layout. What you can now do is pull that text file into a spreadsheet. And we're getting to that.

For now, you have downloaded a file. That's a start. You acquired a database. Congratulations. Lets do some work on it.

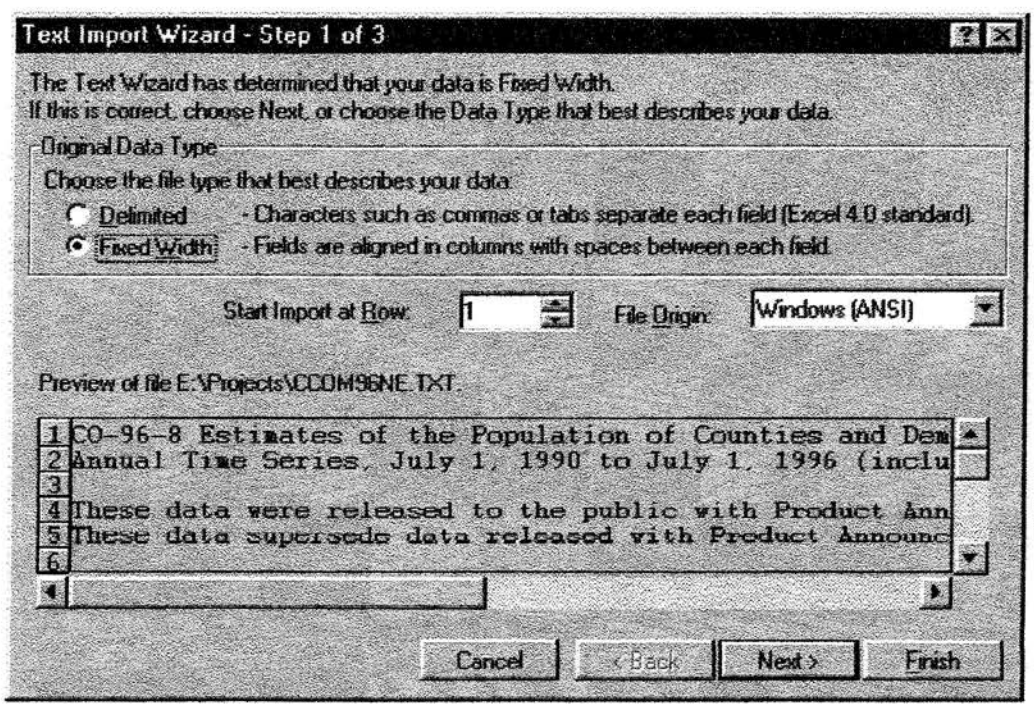
Geeks Covering the Spread: Simple Spreadsheets For Life

The first thing that you need to learn with spreadsheets is importing files into them. It is the first step to anything. Unless you are building your own. In that case, open one and start typing.

But let's go back to our file that we downloaded from the census bureau. If you don't have access to the web, I have included the Nebraska file in the exercise data. It's called CcomNE96.

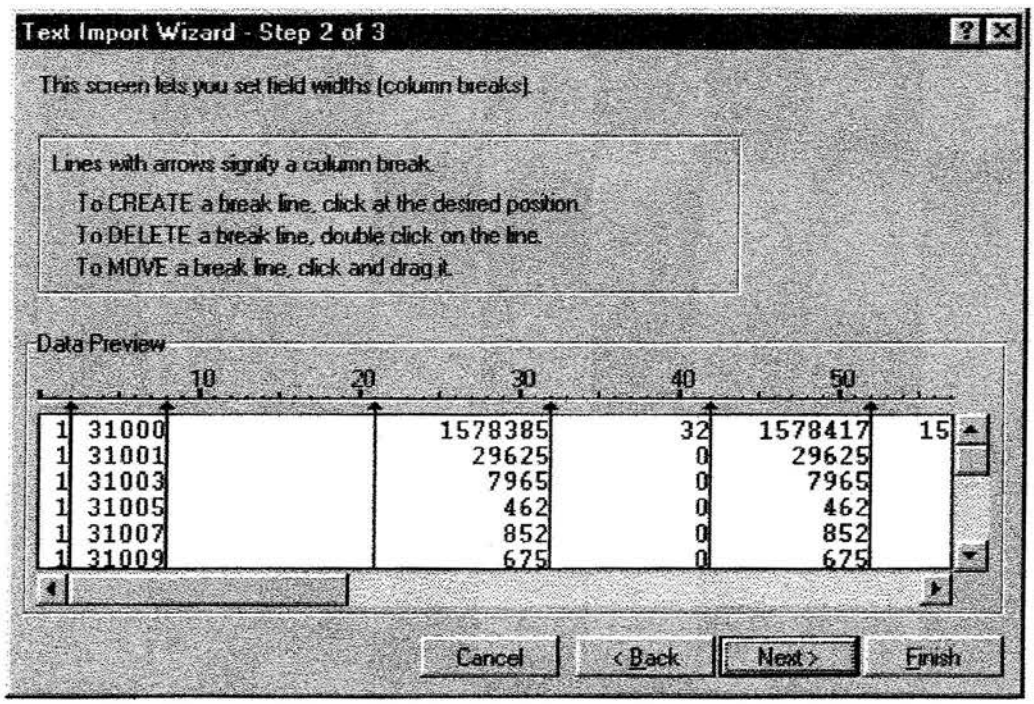
Go ahead and open your spreadsheet program. I am using Excel 97, but what I have seen in other spreadsheets is similar.

Excel has a beautiful thing called the Import Wizard. If you got to open a file, it will give you a screen where it is looking for Excel files. Adjust the dialogue box in the bottom left of the screen to say "all files." Then find your file of census projections. When you click on it to open it, you should get a screen like this.



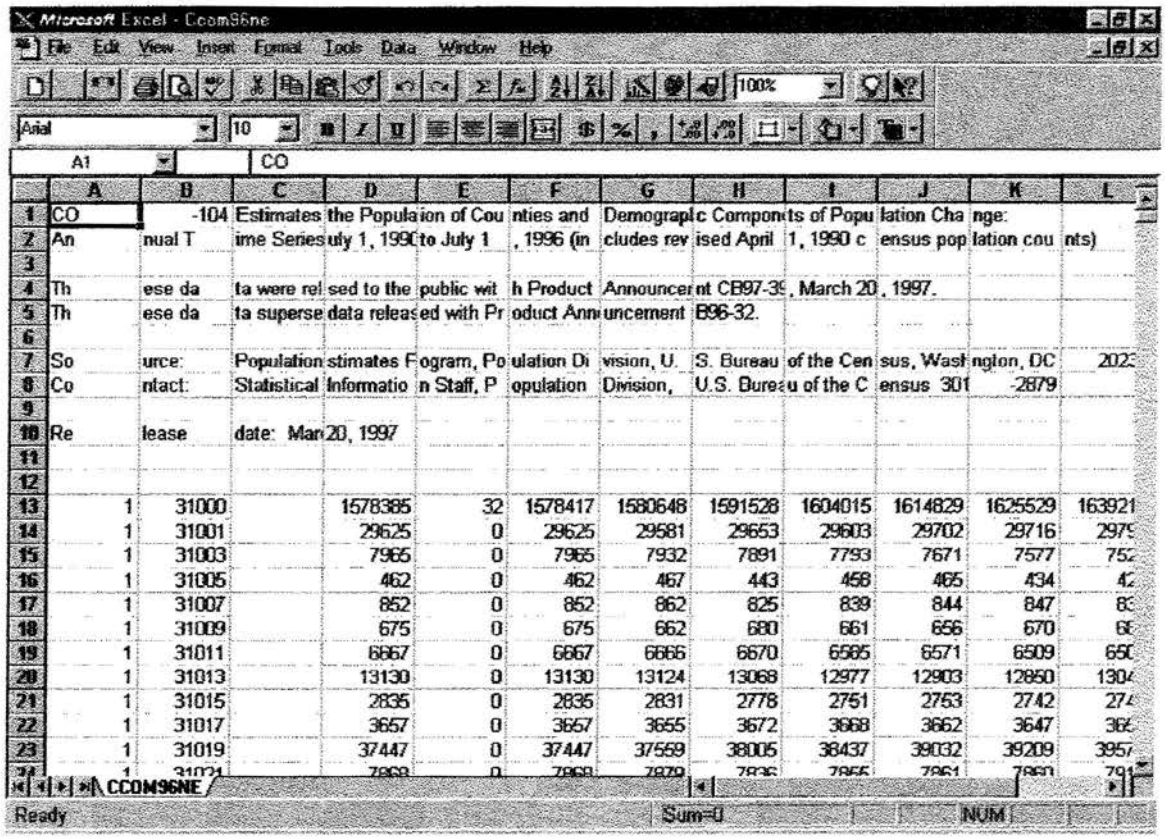
You want it to be a fixed width data type. What that means is you know there are no delimiters, so you are going to assign them with your eye and your mouse.

Go ahead and click next. Since there is that text at the top of your screen, you are going to have to scroll down to where the numbers are. The numbers are the most important here. If you look, you can see that they are organized in columns naturally, and that Excel has already put some columns in for you. Now is when that record layout will come in very handy. That tells you where the lines go. Go ahead and put them in. Generally the best place is to butt them up against the last number in the cell you are going to create. It should look like this.



Go ahead and hit next. Here it asks you to pick what kind of column each should be. General is usually the best, but if you know the column is a date field, then go ahead and make it a date field. If it is a text field (like the last one in our spreadsheet) don't

worry about making it a text file. Excel recognizes it and makes it a text field automatically. Go ahead and click finish. This is what you should get.



	A	B	C	D	E	F	G	H	I	J	K	L
1	CO	-104	Estimates	the Popula	ion of Cou	nties and	Demograph	c Compone	ts of Popu	lation Cha	nge:	
2	An	nual T	ime Series	uly 1, 199	to July 1	, 1996 (in	cludes rev	ised April	1, 1990 c	ensus pop	lation cou	nts)
3												
4	Th	ese da	ta were rel	sed to the	public wit	h Product	Announcer	nt CB97-3	, March 20	, 1997.		
5	Th	ese da	ta superse	data releas	ed with Pr	oduct Ann	uncement	B96-32.				
6												
7	So	urce:	Population	stimates P	ogram, Po	ulation Di	vision, U.	S. Bureau	of the Cen	sus, Wash	ngton, DC	202
8	Co	ntact:	Statistical	Informatio	n Staff, P	opulation	Division,	U.S. Bure	u of the C	ensus 301	-2879	
9												
10	Re	lease	date: Mar	20, 1997								
11												
12												
13	1	31000		1578385	32	1578417	1580648	1591528	1604015	1614829	1625529	163921
14	1	31001		29625	0	29625	29581	29653	29603	29702	29716	297
15	1	31003		7965	0	7965	7932	7891	7793	7671	7577	75
16	1	31005		462	0	462	467	443	458	465	434	4
17	1	31007		852	0	852	862	825	839	844	847	8
18	1	31009		675	0	675	662	680	661	656	670	6
19	1	31011		6667	0	6667	6666	6670	6585	6571	6509	65
20	1	31013		13130	0	13130	13124	13068	12977	12903	12850	130
21	1	31015		2835	0	2835	2831	2778	2751	2753	2742	27
22	1	31017		3657	0	3657	3655	3672	3668	3662	3647	36
23	1	31019		37447	0	37447	37559	38005	38437	39032	39209	395
24	1	31021		7969	0	7969	7979	7936	7955	7961	7960	79

CCOM96NE

Ready Sum=0 NUM

The now-jumbled text can now go. Click on the number at the left side (it should highlight the whole row) and drag down to the 12th row. Punch delete and grab your record layout. It's time to bust this thing up.

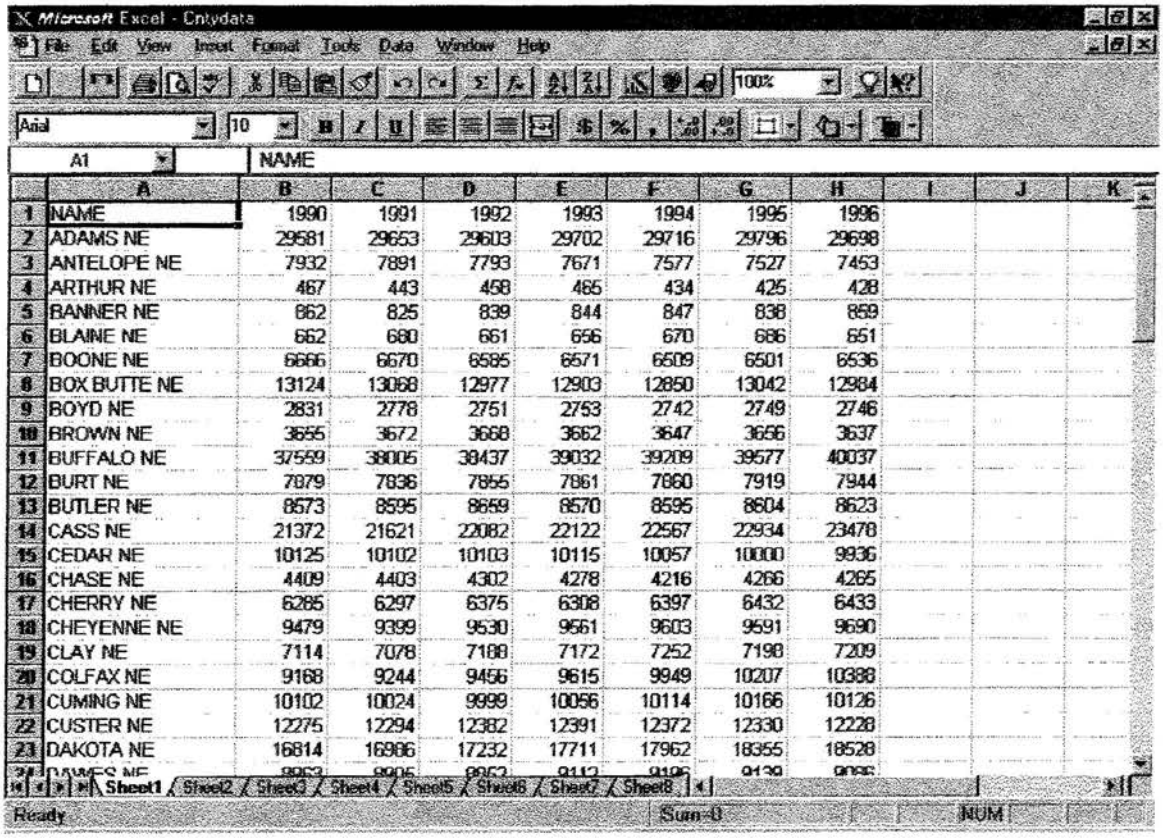
You may have noticed, if you are like me and get curious about these databases and have already scrolled down, that what you have is a series of databases all in one big file. What you are going to have to do is now some trickery. Nothing serious. No jail time. You need to open a new spreadsheet file. When you opened the text file, it opened it as an "active page only" file. That means there is only one sheet to your workbook. With this file, you are going to need many.

Reading the record layout, you can click on the number next to the first record of one database and drag down to the last record. Then go up, copy those cells, and paste them into your new file. Repeat this until you have all of the different databases in the big file broken down in the new workbook file.

And like all things, CAR geeks operate under the "save early, save often" philosophy. So save before you do anything major. And after.

Now that you have all the databases into different sheets, let's start some house keeping. Click on the A, and go up to Insert. Insert a blank column. Now go to the last column of your database. There you will find all the names of your counties. Don't ask me why they put them last. Click on the letter above the county names, cut and paste them into your new field. Then go to 1 and click on it. Insert a blank row, so you can put NAME for your county name field, and all the years your record layout told you about. I would just go ahead and delete the column for census adjustments for 1990. It just bogs things down a bit.

Your sheet should look like this now.



	A	B	C	D	E	F	G	H
1	NAME	1990	1991	1992	1993	1994	1995	1996
2	ADAMS NE	29581	29653	29603	29702	29716	29796	29698
3	ANTELOPE NE	7932	7891	7793	7671	7577	7527	7453
4	ARTHUR NE	467	443	458	485	434	425	428
5	BANNER NE	862	825	839	844	847	838	859
6	BLAINE NE	662	680	661	656	670	686	651
7	BOONE NE	6666	6670	6585	6571	6509	6501	6536
8	BOX BUTTE NE	13124	13068	12977	12903	12850	13042	12984
9	BOYD NE	2831	2778	2751	2753	2742	2749	2746
10	BROWN NE	3655	3672	3668	3662	3647	3656	3637
11	BUFFALO NE	37599	38005	38437	39032	39209	39577	40037
12	BURT NE	7879	7836	7855	7861	7860	7919	7944
13	BUTLER NE	8573	8595	8659	8670	8595	8604	8623
14	CASS NE	21372	21621	22082	22122	22567	22934	23478
15	CEDAR NE	10125	10102	10103	10115	10057	10000	9936
16	CHASE NE	4409	4403	4302	4278	4216	4266	4265
17	CHERRY NE	6285	6297	6375	6308	6397	6432	6433
18	CHEYENNE NE	9479	9399	9530	9561	9603	9591	9690
19	CLAY NE	7114	7078	7189	7172	7252	7198	7209
20	COLFAX NE	9168	9244	9456	9615	9949	10207	10388
21	CUMING NE	10102	10024	9999	10056	10114	10166	10126
22	CUSTER NE	12275	12294	12382	12391	12372	12330	12228
23	DAKOTA NE	16814	16986	17232	17711	17962	18355	18528

Here we have the county, the year of the census projection and the projection itself.

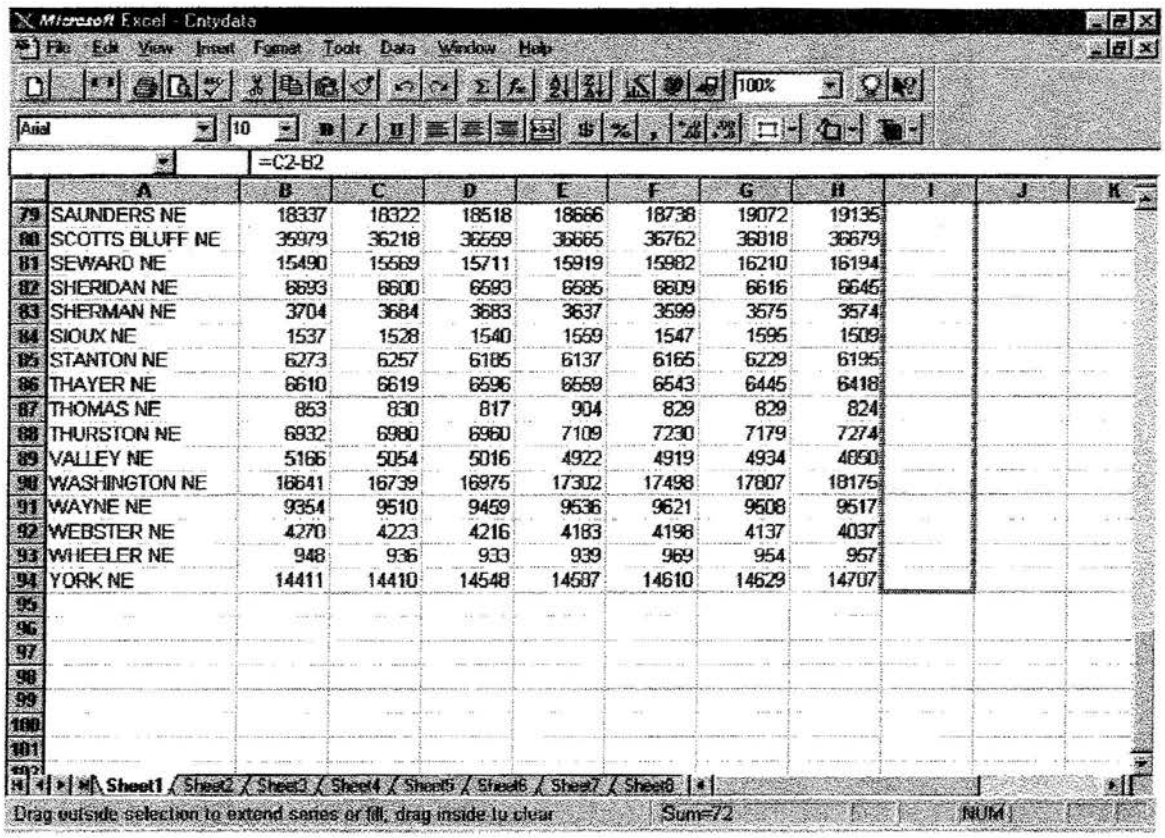
Now the fun begins.

We are now getting into real spreadsheets. What we are going to do is see how much the population changed in each of the years, and in total, for each county. In Sheet 1 of your new file, go over to the next blank row, and start entering in the year intervals – 90-91. 91-92. 92-93. And so on.

Now drop down a line. In order to program a formula in Excel, you need to enter an equal sign into a field. In I2, enter this formula: =C2-B2 and hit enter. Got a number, didn't you. You just told Excel to subtract the 1990 number from the 1991 number, giving you the change in population that year. Neat, huh.

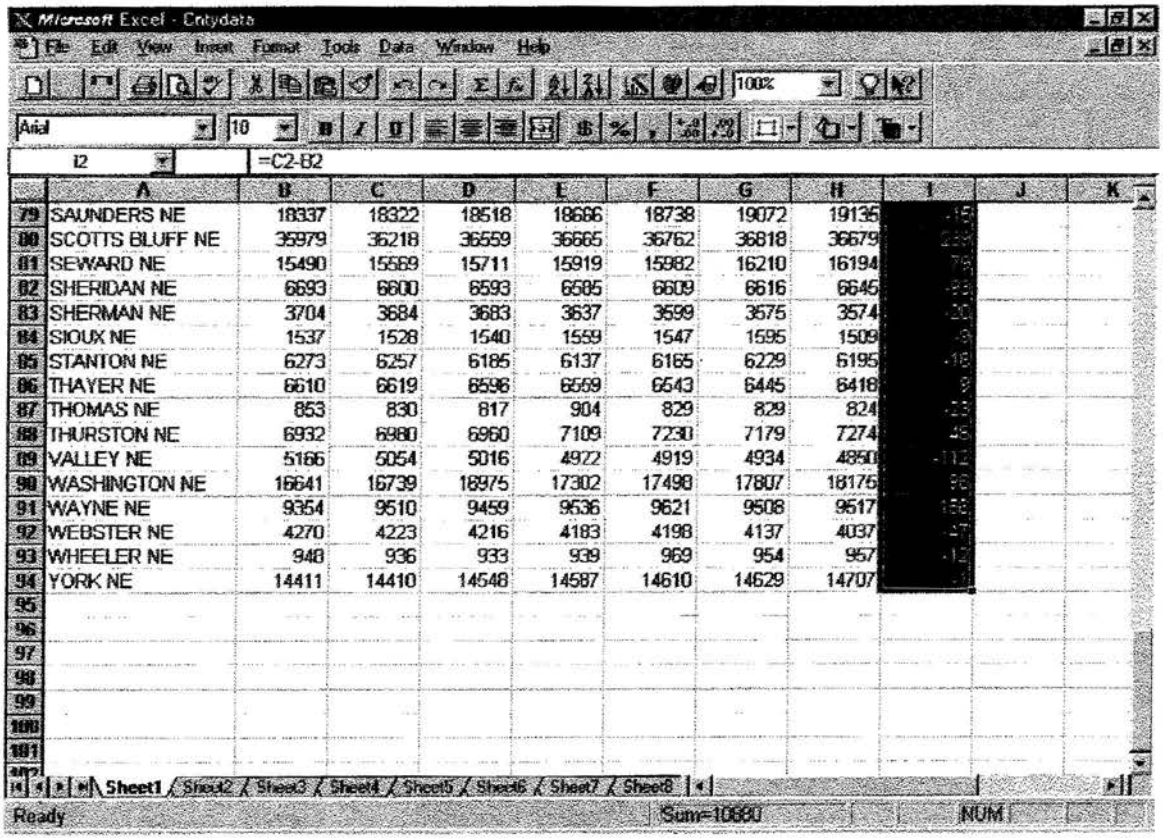
Oh, that ain't nothing yet.

Excel is a great time saver. Do you see that black tag on the bottom right side of the box holding your completed formula? Click on that and hold onto it. Did you see your cursor change? That means you are now going to copy that formula. Drag all the way down the screen to the last county, like this.



	A	B	C	D	E	F	G	H	I	J	K
79	SAUNDERS NE	18337	18322	18518	18686	18738	19072	19135			
80	SCOTTS BLUFF NE	35979	36218	36559	36565	36762	36818	36679			
81	SEWARD NE	15490	15569	15711	15919	15982	16210	16194			
82	SHERIDAN NE	6693	6600	6593	6585	6609	6616	6645			
83	SHERMAN NE	3704	3684	3683	3637	3599	3575	3574			
84	SIOUX NE	1537	1528	1540	1569	1547	1595	1509			
85	STANTON NE	6273	6257	6185	6137	6165	6229	6195			
86	THAYER NE	6610	6619	6596	6559	6543	6445	6418			
87	THOMAS NE	853	830	817	904	829	829	824			
88	THURSTON NE	6932	6980	6960	7109	7230	7179	7274			
89	VALLEY NE	5166	5054	5016	4922	4919	4934	4950			
90	WASHINGTON NE	16641	16739	16975	17302	17498	17807	18175			
91	WAYNE NE	9354	9510	9459	9536	9621	9508	9517			
92	WEBSTER NE	4270	4223	4216	4183	4198	4137	4037			
93	WHEELER NE	948	936	933	939	969	954	957			
94	YORK NE	14411	14410	14548	14587	14610	14629	14707			
95											
96											
97											
98											
99											
100											
101											

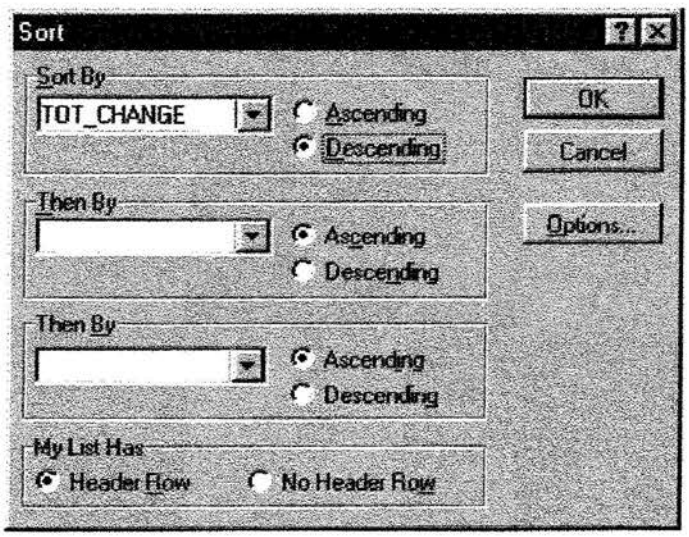
Now let go. It becomes this.



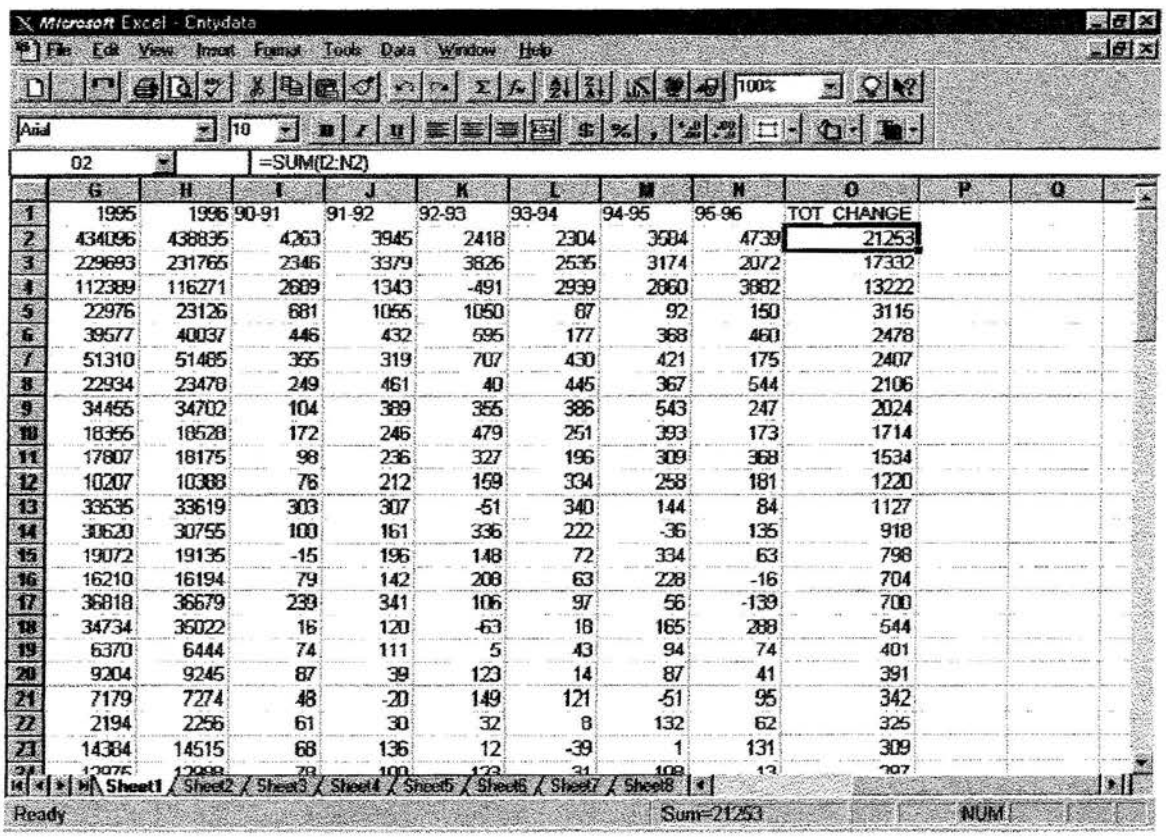
You just saved yourself a heap of work with that. Now go back up to your original formula. Copy the formula across the sheet and release. Then copy the formula for all the other cells in the years.

Now lets try something else. Type in the cell next to your last two year span a field called TOT_CHANGE. Put this formula in: =SUM(I2:N2). What you told it to do was add the fields from I2 to N2 together. Copy that formula down and now you have each counties total change over the last six years.

Now that's nice and all, but editors like things in order. So let's get this in order. In the top left side, there is a box in the corner of the screen. If you click on it, it selects the whole sheet. Do that. And then go to Data, then Sort and release. You should get this screen.



Click on the sort by window, and select the field you want to sort by. In this case, we want TOT_ CHANGE. Then select Descending order, so we can get the fastest growing first, and the fastest shrinking last. Click okay, and voila, you have a list of the fastest growing, and fastest shrinking counties.



	G	H	I	J	K	L	M	N	O	P	Q
1	1995	1996	90-91	91-92	92-93	93-94	94-95	95-96	TOT_CHANGE		
2	434096	438835	4263	3945	2418	2304	3584	4739	21253		
3	229693	231765	2346	3379	3826	2535	3174	2072	17332		
4	112389	116271	2689	1343	-491	2939	2860	3882	13222		
5	22976	23126	681	1055	1050	87	92	150	3115		
6	39577	40037	446	432	595	177	368	460	2478		
7	51310	51485	355	319	707	430	421	175	2407		
8	22934	23478	249	461	40	445	367	544	2106		
9	34455	34702	104	389	355	386	543	247	2024		
10	18355	18528	172	246	479	251	393	173	1714		
11	17807	18175	98	236	327	196	309	368	1534		
12	10207	10388	76	212	159	334	258	181	1220		
13	33535	33619	303	307	-51	340	144	84	1127		
14	30620	30755	100	161	336	222	-36	135	918		
15	19072	19135	-15	196	148	72	334	63	798		
16	16210	16194	79	142	208	63	228	-16	704		
17	36818	36679	239	341	106	97	56	-139	700		
18	34734	35022	16	120	-63	18	165	288	544		
19	6370	6444	74	111	5	43	94	74	401		
20	9204	9245	87	39	123	14	87	41	391		
21	7179	7274	48	-20	149	121	-51	95	342		
22	2194	2256	61	30	32	8	132	62	325		
23	14384	14515	68	136	12	-39	1	131	309		

In other words, you used a spreadsheet for CAR.

Remember the story about the first day at the Arkansas Democrat-Gazette? I did a spreadsheet exactly like that with county body dispositions for 10 years. Every Arkansas who died, I had what happened to their body. I wrote two simple formulas, did a sort, and had a spreadsheet that lead me to my first 1A story for that paper.

Spreadsheets are super-powerful and super easy to use. The more you play with them, the more you can do with them. For beat writers, spreadsheets are great because they can enter last year's budget for their agency and this year's budget, do some simple formula's and figure out who is getting the biggest increase. Spreadsheets can do averages (=AVERAGE (A2:A7), etc. etc.) and medians and even really complex statistics. It just all depends on what you want to do with it. Spreadsheets are the most useful, and often used, tools of CAR. Learn them, and half of CAR is yours.

Heavy Geek Lifting: Doin' the Dirty Work with Databases

So you may be asking yourself, "If spreadsheets do all that, why do I need databases?" Well, quite simply, it's like football. Spreadsheets are your skill players – your wide receivers, running backs and quarterbacks. They do all the dancing and running. Databases are all about the offensive line. They are the muscle. The real muscle. They do the heavy work. More technically, spreadsheets are only good for so many thousands of records (16,000 for Excel) and some databases have millions of records. Without databases, and the knowledge of how you use them, you may have a database of great information, and nothing to do with it.

Databases are such heavy lifters that they get their own language. Most commonly, the language is called SQL, or structured query language. Now that's really just one language, but most of them are very similar, so they get the blanket term SQL. I will be honest with you. I don't know much SQL. There are a large number of CAR geeks who don't. Others do, and swear by it. If you want to learn all of it, more power to you. We're going to do some SQL queries, because following the query wizards you can do on your own. They are pretty self explanatory. SQL gets a little trickier, but not much.

The two databases that I have done the most work in a database manager have been campaign finance information and crime data. Campaign finance data is out there for the taking, from various websites and campaign finance stories are some of the easiest, and most common, CAR stories out there. Editors like them, people are interested in them, and with a few clicks, the CAR work is easily done.

Tip sheets and campaign finance story instructions are out there for the taking (the NICAR website is now running a campaign finance center at http://www.campaignfinance.org/ Other organizations have tips for you also. Scroll through the bookmarks included in the data and you will find several with good information.

Since they are more experienced at campaign finance than I, I'll leave you to their expert tutelage on those issues. I'll stick to crime, my personal favorite.

In February 1996, I was starting to get into CAR, and I wanted to do crime stories. So I started asking about databases at the Lincoln Police Department. What I learned stunned me. They had so many databases it was mesmerizing. So, asking nicely, I asked the police chief for the 1995 incidents file. After some sweet-talking and dealing, he gave me the whole file for free. I took it, broke it down, cleaned it up, mapped the crimes and did neighborhood by neighborhood crime rates.

But there are much easier stories to do from that file. What I did was glean out certain crimes that tend to make the news around here. No, there was no homicide file. We had one in 1995. I had things like assaults, weapons offenses, drugs, car thefts, larcenies, disturbances, robberies and burglaries. Larcenies took up a full one-fourth of the database. There were more than 10,000 larcenies in 1995 and just short of 40,000 total incidents.

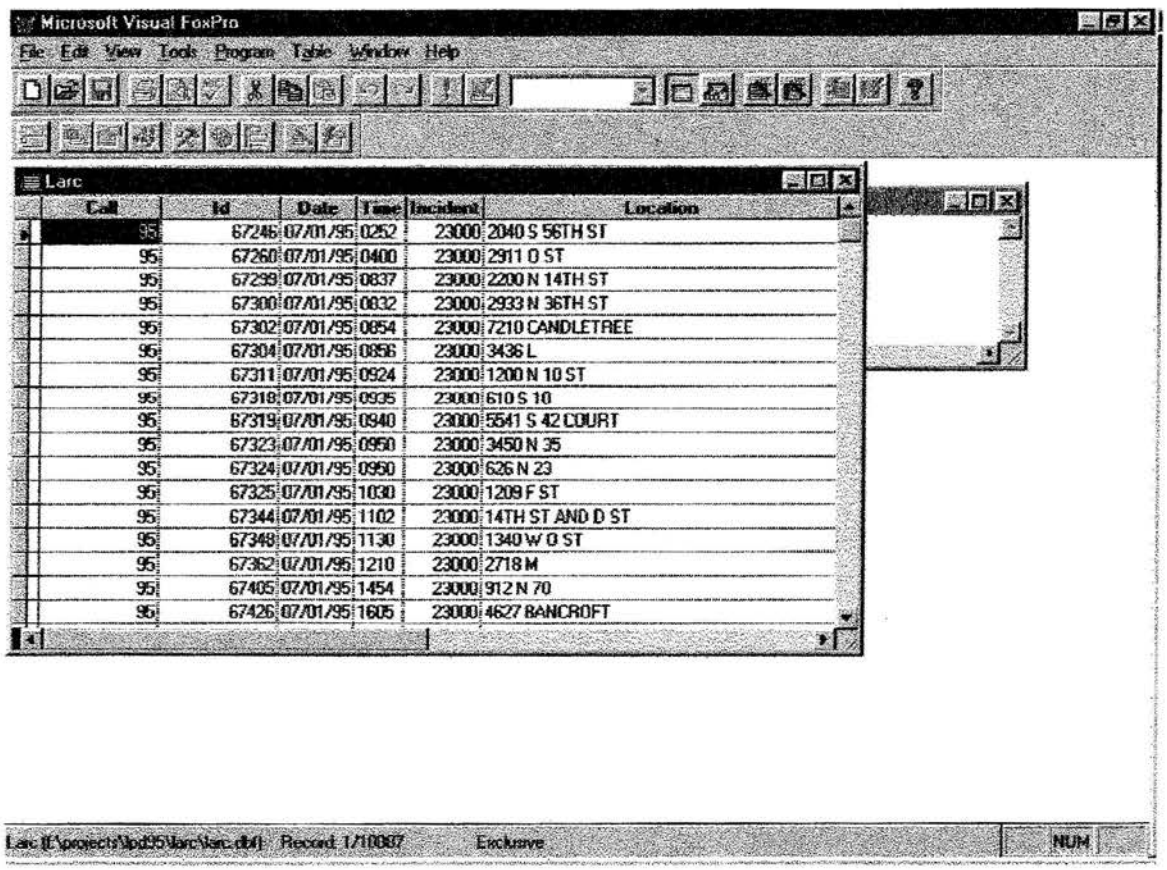
Let's look at that larceny database. Let's try and figure out what day did the police receive the most larceny reports, or, what day did the most people lose stuff.

I am using Visual FoxPro 5.0. I used to use Fox 2.6. Others swear by Microsoft Access. I have that too, but I don't know how to use it. If you are just getting into CAR,

chances are you have Access or Visual FoxPro. In CAR, this is the Ford/Chevy debate – the Coke or Pepsi question. Which is better? That's up to you. I'm not getting into that dogfight.

In Visual FoxPro, go to FILE, OPEN, and then find the folder you have the exercise data in. Change the file type window to table (.dbf) so it will show LARC.dbf. This is a database of all larcenies in Lincoln, Nebraska in 1995.

Just because you did that doesn't mean you told FoxPro that you wanted to see it. Database programs are like children: you have to tell them everything they are supposed to do. In the Command window, type BROWSE and hit enter. You should get this:



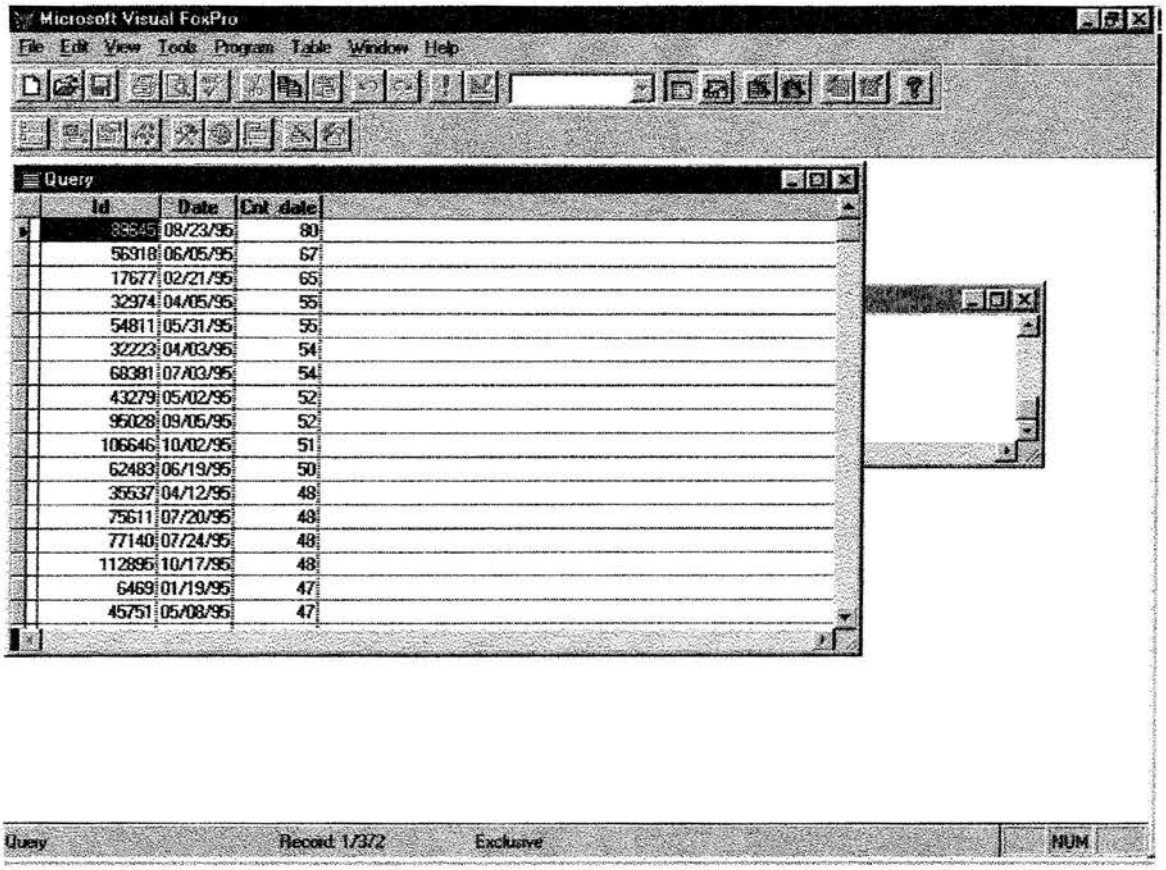
Here, we are going to learn some simple SQL. In SQL, certain words tell FoxPro to do certain things. First, you want to select. Select tells the computer you want a certain file.

In SQL, it is important to remember that if you do not have a semi-colon at the end of the command, it tells the computer to do it. So end every phrase with a semicolon until you want to start it.

So in the command window, type "SELECT larc.id, larc.date, COUNT (larc.date);" That way, you have told the computer to select the id field from larc, date from larc, and to count all the dates. Now tell the computer where to select the fields from. Type "FROM larc;"

Now, we need to tell the computer to lump all the similar fields together. So all the January 1s are with each other, and so on and so forth. You do that with the Group By command. Now type "GROUP BY Larc.date;" And now, tell the computer what order you want it in by typing "ORDER BY 3 DESC;" What you just told it was I want you to order the database by the third column in descending order. You could go the other way, and say ASC, but for most journalistic purposes, DESC is the way to go. Biggest up top, lowest down at the bottom.

Now, the last thing you need to do is type BROWSE. This is what you should get:



As you can see, Aug. 23 is by far the most larcenous day in Lincoln.

But wait. Look down below. See the Record: 1/372. That is your location in the database. You are at record 1 of 372.

Doesn't that seem odd to you?

Since when have their been 372 days in a year?

This is where double checking everything comes in handy. Scroll down to the bottom. There you will find a series of records that have incorrect date fields. And it will have their ID number, so you could get the paper records for your story.

Keep your eyes open.

Other Toys for Other Times

Now that we have used the meat and potatoes of CAR, I just want to take a second to show you one of the toys. I call them that, anyway. They are really powerful tools that are nice to have (and expensive) but not necessary to do CAR.

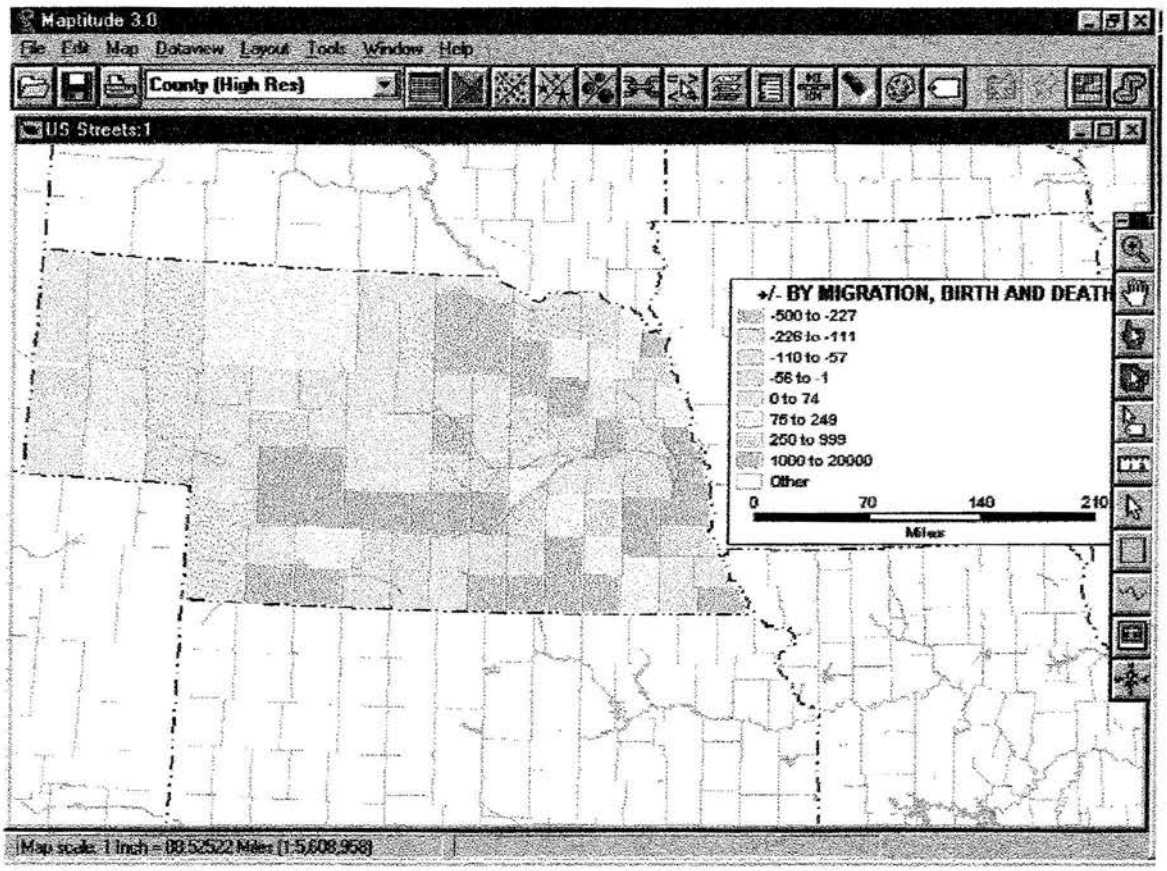
Remember the spreadsheet we made of the population changes? What if you could make a map out of that data? You can.

There are a group of programs call GIS, or Global Information Systems. GIS software, like Atlas GIS or Mapinfo, are really powerful, really expensive programs. Maptitude, the program I use, isn't as powerful, but is drastically cheaper (were talking four digits). Maptitude is easy to learn and fun to master.

With Maptitude, I took that spreadsheet and made some new tables. In those, I took all the births, deaths and the net migration numbers and threw them all together. That way, I had the population change due to those three things, which accounts for the majority in all of the counties in Nebraska. I took that, made the county names match the county names in Maptitude's database of census information.

With those two fields matching, I could then link it to the program's database of census information. The Caliper Corp. has made it really easy to link your own data to theirs, and then turn it into maps you want to make. With the click of a button, I created a thematic map.

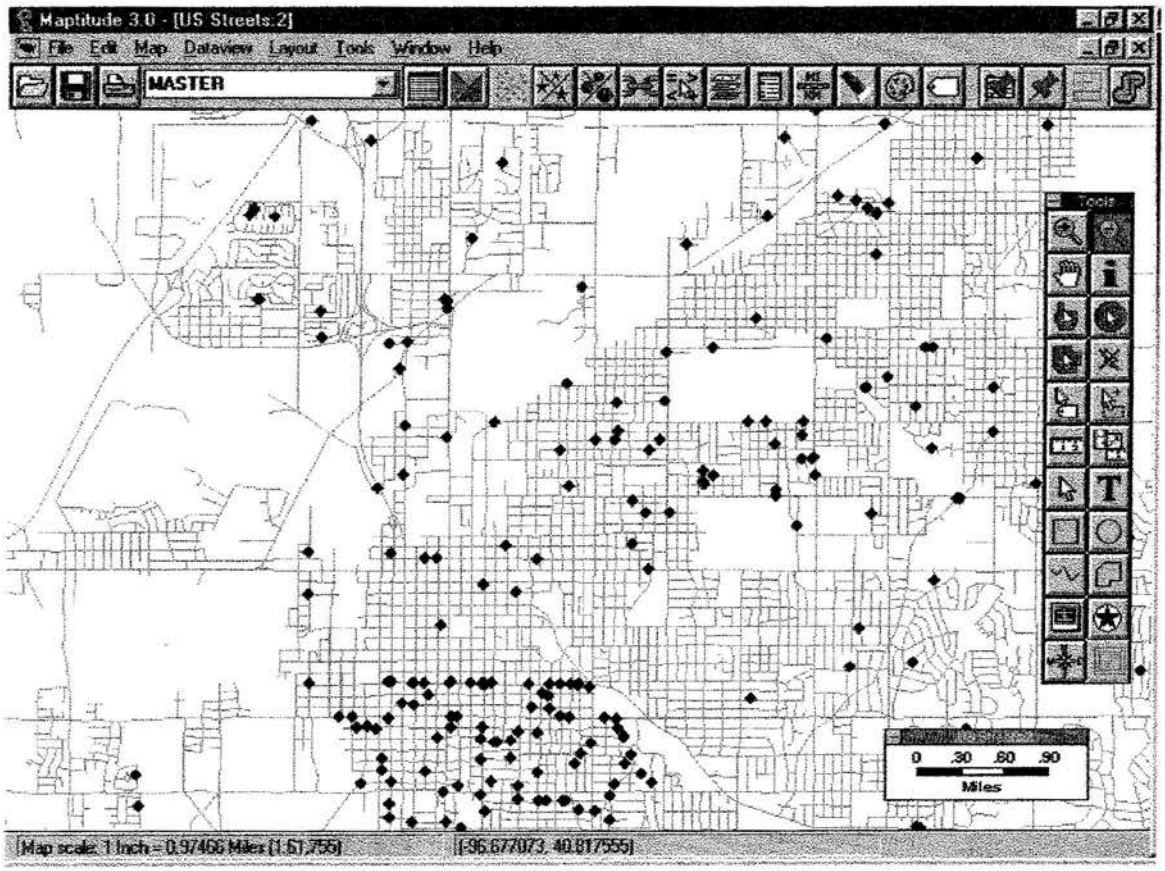
It looked like this:



Thematic mapping presents your data in color themes. That way, you can just look at the map and see trends.

There are other kinds of mapping also. One of the most powerful is pin mapping. Pin mapping is when the computer reads an address in your database, finds it on the city map, and then puts a point there.

For example, here's a map of all property fires for an area of Lincoln.



The power of pin mapping was made most clear to me when I was working on the neighborhood by neighborhood crime rates. Using pin mapping, I showed that a neighborhood that the whole city thought was a high-crime area was really not. The story help start a revival in the neighborhood, changed police procedures, unleashed millions in community development funds and restored pride to long-time residents. It was all because of Maptitude and willpower.

There are other toys out there for the finding. New toys for the making. Keep your ear to the ground and your mind open to learning new programs and you will stay ahead of the curve. And you will keep CAR alive.

The End of the Road

So now, if any of this has sunk in, you have taken your first steps towards being a geek like me. I hope you now see that CAR is not about high-power computers, high-fallooting programming language, gibberish that no one understands, statistics that are only understood in math departments and general nausea to a journalist.

CAR is just a tool. It is no substitute for good old fashioned reporting. That's were the order of things come in. You need to find the people with the data, get the data, work the data, find the trends, then go find the people in the trends. These numbers mean nothing to a reader without people. Only academics read page after page of numbers and transitions.

What CAR offers is the ability to make a story deeper with a more solid context. You can say that the northeastern part of the state is growing at a rapid rate, but with CAR, you can tell your readers exactly how fast, and exactly where in the northeastern part of the state. CAR gives you a level of detail that just wasn't available before. Now, editors are clamoring for it.

Hopefully I have brought some of the mental walls down, and showed you that being a geek isn't all bad. Like I said earlier, when people call me a geek, I don't mind, because I know my story is going to be better because of it.

And if that makes me a geek, call me guilty.

How about you?