SPREADSHEETS

Submitted for Publication in the Encyclopedia of Information Systems
Publisher: Academic Press

By
Daniel J. Power
Department of Management
College of Business Administration
University of Northern Iowa
Cedar falls, IA 50614-0125
Phone: (319)-273-2987
e-mail: power@uni.edu

And
Shashidhar Kaparthi
Department of Management
College of Business Administration
University of Northern Iowa
Cedar falls, IA 50614-0125
Phone: (319)-273-2958
e-mail: kaparthi@uni.edu

August 1, 2000

OUTLINE

Glossary

Absolute Reference – A cell reference in a formula whose location remains the same when copied.

Cell – The position created by the intersection of a column and row.

Goal seeking - The capability of asking the computer software model what values certain variables must have in order to attain desired goals. It is a tool that uses iterative calculations to find the value required in one cell (variable) in order to achieve a desired value in another cell.

Instant Recalculation – The recalculation of formulas whenever values in referenced cells change.

Relative Reference – A cell reference in a formula whose location adjusts when copied.

Spreadsheet - a spreadsheet is a program that allows any part of a rectangular array of positions or cells to be displayed on a computer screen, with the contents of any cell able to be specified independently or in terms of the contents of other cells.

What-If Analysis - The capability of "asking" the software package what the effect will be of changing some of the input data or independent variables.

I. Introduction

In the realm of accounting jargon a "spread sheet" or spreadsheet was and is a large sheet of paper with columns and rows that lays everything out about transactions for a businessperson to examine. It spreads or shows all of the costs, income or taxes on a single sheet of paper for a manager to look at when making a decision. An electronic spreadsheet organizes information into software-defined columns and rows. Data can then be "added up" by a formula to give a total or sum. A spreadsheet program summarizes information from many paper sources in one place and presents the information in a format to help a decision-maker see the financial "big picture" for a company. In general, the term spreadsheet is used by Information Systems professionals to categorize programs that allow any part of a rectangular array of positions or cells to be displayed on a computer screen, with the contents of any cell able to be specified independently or in terms of the contents of other cells.

Based on sales figures millions of managers, business analysts and professionals around the world create spreadsheets each year. Creating spreadsheets is important in organizations and many mission-critical corporate decisions are guided by the results of large and complex spreadsheets. This article is a review, a discussion and an exposition of spreadsheets. A brief history of spreadsheets is provided in the next section followed by a review of spreadsheet fundamentals and concepts. Then, spreadsheet applications are discussed, followed by a review of some academic research in this area.

II. Brief History

Spreadsheet history begins hundreds of years ago, but electronic spreadsheets are of much more recent origin. Some published newspaper and magazine stories have reported that in 1978 a Harvard Business School student, Daniel Bricklin, invented the first electronic spreadsheet called VisiCalc. Bricklin came up with the idea for an interactive visible calculator. Bricklin considers Bob Frankston a co-inventor of VisiCalc.

The VisiCalc spreadsheet program was the first major application for personal computers.

In the early 1960s, Professor Richard Mattessich pioneered computerized spreadsheets for business accounting in a paper published in The Accounting Review. Some historical information on the computerization of accounting spreadsheets is discussed on Mattessich's web page "Spreadsheet: Its First Computerization (1961-1964)" at URL http://www.j-walk.com/ss/history/spreadsh.htm. Mattessich's work influenced the computerization of spreadsheets programmed on mainframe computers, but it is unlikely that it influenced Bricklin and Frankston. Although some spreadsheet development occurred prior to the invention of VisiCalc, it began the modern spreadsheet era.

The tale of VisiCalc is part myth and part fact for most of us. The story is that Dan Bricklin was preparing a spread sheet analysis for a Harvard Business School "case study" report and had two alternatives: 1) do it by hand; or 2) use a clumsy time-sharing mainframe program. Bricklin thought there must be a better way. He wanted a program where people could visualize the spreadsheet as they created it. His metaphor was "an electronic blackboard and electronic chalk in a classroom." By the fall of 1978, Bricklin had programmed the first working version of his concept. The program would let users input a matrix of five columns and 20 rows. The first version was not very "user friendly" so Bricklin recruited an MIT acquaintance Bob Frankston to improve and expand the program. Frankston expanded the program and "packed the code into a mere 20k of machine memory, making it both powerful and practical enough to be run on a microcomputer". Dan Bricklin has his version of the history of Software Arts and VisiCalc on the web at www.bricklin.com/history/sai.htm.

During late fall of 1978, Daniel Fylstra, founding Associate Editor of Byte Magazine, joined Bricklin and Frankston in developing VisiCalc. Fylstra was also an MIT/HBS

graduate. Fylstra was "marketing-oriented" and suggested that the product would be viable if it could run on an Apple microcomputer. Bricklin and Frankston formed Software Arts Corporation on January 2, 1979. In May 1979, Fylstra and his firm, Personal Software (later renamed VisiCorp), began marketing "VisiCalc" with a pre-release ad in Byte Magazine. The name "VisiCalc" is a compressed form of the phrase "visible calculator". VisiCalc became an almost instant success and provided many business people with an incentive to purchase a personal computer. It was even available on an H-P 85 or 87 calculators from Hewlett-Packard. About 500,000 copies of the spreadsheet program were sold during VisiCalc's product lifetime. A screen shot of the first version of VisiCalc is provided in Figure 1.

Insert Figure 1 Here

A. What came after VisiCalc?

The market for electronic spreadsheet software was growing rapidly in the early 1980s and VisiCalc was slow to respond to the introduction of the IBM PC that used an Intel computer chip. Beginning in September 1983, legal conflicts between VisiCorp and Software Arts distracted the VisiCalc developers. During this period, Mitch Kapor developed Lotus and his spreadsheet program quickly became the new industry spreadsheet standard.

B. Lotus 1-2-3

Lotus 1-2-3 made it easier to use spreadsheets and added integrated charting, plotting and database capabilities. Lotus 1-2-3 established spreadsheet software as a major data presentation package as well as a complex calculation tool. Lotus was also the first spreadsheet vendor to introduce naming cells, cell ranges and spreadsheet macros. Kapor was the VisiCalc product manager at Personal Software for about six months in 1980; he also designed and programmed Visiplot/Visitrend that he sold to Personal Software (VisiCorp) for $1 million. Part of that money along with funds from venture capitalist Ben Rosen was used to start Lotus Development Corporation in 1982. Kapor

co-founded Lotus Development Corporation with Jonathan Sachs. Before he struck out on his own, Kapor offered to sell Personal Software (VisiCorp) his initial Lotus program. Supposedly VisiCorp executives declined the offer because Lotus 1-2-3's functionality was "too limited". Lotus 1-2-3 is still one of the all-time best selling application software packages in the world.

In 1985 Lotus acquired Software Arts and discontinued VisiCalc. A Lotus spokesperson indicated at that time "1-2-3 and Symphony are much better products so VisiCalc is no longer necessary."

C. Microsoft Excel

The next milestone was the Microsoft Excel spreadsheet. Excel was originally written for the 512K Apple Macintosh in 1984-1985. It was one of the first spreadsheets to use a graphical interface with pull down menus and a point and click capability using a mouse-pointing device. With its graphical user interface, Excel was easier for most people to use than the command line interface of PC-DOS spreadsheet products. Many people bought Apple Macintoshes so that they could use the Excel spreadsheet.

When Microsoft launched the Windows operating system in 1987, Excel was one of the first products to be released for it. When Windows finally gained wide acceptance with Version 3.0 in late 1989, Excel was Microsoft's flagship product. Excel remained the only Windows spreadsheet program for nearly 3 years and lacked significant competition from other spreadsheet products until the summer of 1992. In the late 1980s, many companies had introduced DOS spreadsheet products. Spreadsheet products and the spreadsheet software industry were maturing. Microsoft had joined the fray with its innovative Excel spreadsheet. Lotus had acquired Software Arts and the rights to VisiCalc. Jim Manzi had become CEO at Lotus in April 1986 and in July 1986 Mitch Kapor resigned as Chairman of the Board. The entrepreneurs were moving on ...

D. Legal Battles

In January of 1987, Lotus Development filed suit against Paperback Software and separately against Mosaic Software claiming they had infringed on the Lotus 1-2-3-spreadsheet software. In a related matter, Software Arts, the developer of the original VisiCalc spreadsheet software filed a separate action against Lotus claiming that Lotus 1-2-3 was an infringement of VisiCalc. Briefly, Lotus won the legal battles, but lost the "market share war" to Microsoft. According to Russo and Nafziger (1993) "The Court granted Lotus' motion dismissing the Software Arts' action and confirming that Lotus had acquired all rights, including all claims, as part of the earlier transaction." Most people have probably forgotten the Lotus clones called TWIN and VP Planner. TWIN and VP Planner were designed to work like Lotus' 1-2-3. Advertising proclaimed that the TWIN software product "offers you so much more, for so much less."

Russo and Nafziger note "Both Mosaic's TWIN and Paperback's VP Planner had most of the same features, commands, macro language, syntax, organization and sequence of menus and messages as Lotus' 1-2-3. Their visual displays were not however identical to 1-2-3 or to each other. Both TWIN and VP Planner reorganized and placed their respective menus, sub-menus, prompts and messages on the bottom of the screen."

On June 28, 1990, Judge Keeton of the Federal District Court in Boston upheld the copyright of the Lotus 1-2-3-user interface. The Court ruled that "[t]his particular expression of a menu structure is not essential to the electronic spreadsheet idea, nor does it merge with the somewhat less abstract idea of a menu structure for an electronic spreadsheet.... the overall structure, the order of commands in each menu line, the choice of letters, words, or 'symbolic tokens' to represent each command, the presentation of these symbolic tokens on the screen, the type of menu system used, and the long prompts -- could be expressed in a great many if not literally unlimited number of ways." Lotus Dev. Corp. v. Paperback Software Int'l, 740 F.Supp. 37, 67 (D.Mass. 1990). Essentially, the court stated that the developers of TWIN and VP

Planner products had other design choices available to them, but chose instead to inappropriately copy the Lotus 1-2-3 interface. In the late spring of 1995, IBM acquired Lotus Development and Microsoft Excel had become the spreadsheet market leader.

III. Fundamentals and Capabilities

Spreadsheet software allows users to manipulate workbooks. A spreadsheet workbook can contain one or more worksheets. A worksheet contains many rows and columns. A hierarchical overview of spreadsheet concepts is presented in Figure 2.

Insert Figure 2 Here

Typically, the rows in the worksheet are numbered sequentially (1, 2, 3, 4, etc.) and the columns in the worksheet are identified by the letters of the alphabet (A, B, C, ... Z, AA, AB etc.) as shown in Figure 3. The rectangular elements formed by the intersection of columns and rows are known as cells. Each cell is unique and can be identified by its 'address' or its 'reference' that is the letter of its column followed by the number of its row. For example, in Figure 2, the address of the highlighted cell is C5, since Column C and Row 5 intersect to form the cell.

Insert Figure 3 Here

A. Cell Contents

A user can type text or labels into cells; a user can type numbers or values into cells; or more importantly, a user can type formulas into cells. A formula is a mathematical expression that may contain numbers, cell addresses, mathematical operators (+: addition, -: subtraction, *: multiplication, /: division, ^: exponentiation) and parentheses. When a user types a formula in a cell, the spreadsheet software computes the formula by substituting the values contained in the cells referenced in the formula and displays the result of the computation in that cell. Figure 4 illustrates this by using an example of a car loan analysis done in Microsoft Excel. In the example, the cost of the car is entered in cell C3, the down payment is entered in cell C4 and a formula

(=C3-C4) is entered in cell C6. The spreadsheet software computes the formula by substituting the values contained in the cells referenced (i.e. 28000 for C3 and 4000 for C4) and displays the result of the computation (28000 - 4000 = 24000) in the cell C6.

Insert Figure 4 Here

B. Built-in Functions

To complete the car loan analysis and to compute the monthly payment, the user needs to perform a complex calculation that takes into consideration the annual interest rate as well as the time period for paying back the loan. Typically spreadsheet software have a capability for performing complex calculations through the use of built-in functions or @ functions (Lotus 123 terminology). Built-in functions process one or more values known are arguments or parameters and output a single resultant value similar to the definition of a function in mathematics. Spreadsheets provide functions for financial, date and time, math and trigonometry, statistical, lookup and reference, database, and engineering computations. Typically, built-in help features are a good source of information on the different kinds of functions available. For the monthly payment to be computed, the spreadsheet has to process the loan required, the interest rate and the time period for paying back the loan. In the example illustrated in Figure 3, a built-in function is entered in cell C11 (=PMT(C8/12,C9,C6)) for computing the monthly payment. The annual interest rate is in C8, the time period in months is in C9 and the loan required is in C6. The result of the computation performed by the PMT built-in function is displayed in C11.

C. Cell Formatting

A user can format cells to better convey the significance of the contents to the audience. In the car loan analysis example (Figure 3), the dollar amounts have been formatted using the 'Currency' format and the cell containing the interest rate has been formatted using the 'Percentage' format. Some of the other formats available include the text, the date, and user-defined formats. To enhance the appearance of the

worksheet a user can format the borders, patterns, colors and fonts of cells. Column widths may be adjusted, row height may be changed and the text can be aligned horizontally as well as vertically as needed in the cells. Modern spreadsheet software even have the capability of orienting the text in the cells at any angle.

D. Instant Recalculation

Returning to the car loan example, if the user wants to reduce the monthly payment by putting more money down ($6,000 instead of $4,000), all the user needs to do is to type the new value into the cell C4. As soon as the user types the new value ($6,000) into cell C4, the spreadsheet software changes the loan required to $22,000 and the monthly payment drops to $526.83. Whenever a value is changed, spreadsheet software immediately recalculates and changes the values of all the formulas that depend on this value. This is a very important feature of spreadsheet software and is known as the 'Instant Recalculation' feature.

Even though the formulas have been setup using one particular set of values, values can be changed and the results obtained for any new situation that a user wants to examine. To put it in simpler terms, once a worksheet has been setup to analyze a particular car loan it can be used to analyze any car loan. Hence, we have a 'model' of a decision-making situation instead of just calculations for one particular decision-making situation. If we examine the car loan example in terms of a mathematical model, we have independent variables (cost of the car, down payment, time period), intermediate variables (loan required), uncontrollable variables (interest rate) and dependent variables (monthly payment). Spreadsheet software allow users to establish mathematical relationships between all of these variables.

E. What-if Analysis?

Once a mathematical model is built a user can very quickly find out the impact of changes in independent variables on the dependent variables and can make an 'informed' decision. This is possible due to the instant recalculation capability of the

spreadsheet software. In the car loan example a user could easily determine the answer to questions like: What is the impact of increasing the down payment by $2,000? What is reduction in the monthly payment if the loan is paid back in 5 years instead of 4 years? What is the impact of cutting the cost by going with fewer options? An analysis that allows us to change the independent variables and examine the impact on dependent variables is known in general as a what-if analysis. For example, a marketing executive should be able to answer the question, what is the impact of an increase in advertising expenses of 20% on sales? To summarize, what-if analysis is the capability that allows users to "ask" the software package what the effect will be of changing some of the input data or independent variables in a model.

Closely related to what-if analysis is the concept of scenario management. A scenario is a set of values that a spreadsheet saves and can substitute automatically in a worksheet. Users can use scenarios to forecast the outcome of a worksheet model. Users can create and save different groups of values on a worksheet and then switch to any of these new scenarios to view different results. Users can compare the results of several scenarios side-by-side and make informed decisions.

F. Goal Seeking

Goal Seeking allows a user to calculate the value of an independent variable needed to achieve a certain value or goal of a dependent variable. It is a tool that uses iterative calculations to find the value required in one cell (variable) in order to achieve a desired value in another cell. A common use of the goal-seeking feature in a spreadsheet is calculating a break-even quantity.

Examples of questions to which a goal seeking capability would give us answers are: What should the cost of the car be for the monthly payment to be $500.00? What should I spend on advertising to realize sales of $50,000 this quarter?

G. Relative and Absolute Addressing

Consider the car loan analysis again. The user wants to compare two car loans side by side to get a better idea of the trade-off involved. Instead of retyping the entire model, the user may use spreadsheet software's copy operation. This allows the user to make a copy of a series of cells and place a duplicate of those cells in another area of the spreadsheet. Copying is generally a two-step process. First, a user highlights (selects) the data that should be copied and then picks the appropriate items from the menu for copying the data over to the clipboard (computer's memory). After that, the user selects the cell(s) that are to be the destination for the data and then picks the appropriate menu items for pasting the data from the clipboard. Figure 5 illustrates the car loan example after the user has copied over the data to column D.

Insert Figure 5 Here

If the second car costs $30,000, all the user needs to do is to type in the new cost in the cell for the cost of the new car (D3). As soon as the user enters the value of 30000 into cell D3, the loan required in cell D6 and the monthly payment in the cell D11 are instantly recalculated as $26,000.00 and $622.60. Even though the user had initially designed the model in column C, when the cell contents are copied over to column D, a model independent from the initial model is copied into column D. The dependent variable in column D depends upon the independent variables and the intermediate variables in column D but not column C. If one examines the formula in cell D6, it is now actually (=D3-D4), even though the original formula copied was =C3-C4. The spreadsheet software has modified the formula when copying, according to the concept of 'relative addressing'.

For the purposes of copying, the concept of relative addressing interprets a formula in terms of the relative positioning of the cells referenced in the formula with respect to the cell containing the formula. This relative positioning is maintained when the formula is copied into the new cell. Obviously, this feature is very important for increasing the

productivity with which a user can build a complex worksheet. In certain situations a user may want a formula to contain a cell reference that should not change when the formula is copied to other cells. A user can do this by using absolute cell references. Figure 6 illustrates examples of formulas containing relative addressing and absolute addressing and resulting formulas when these are copied to other cells.

Insert Figure 6 Here

H. Data Visualization and Decision Support

Modern spreadsheets make use of graphical user interfaces and allow users to perform sophisticated data visualization. Data visualization is an important technique for gathering "information" from data and assisting decision-making. Some of the data visualization techniques that modern spreadsheets allow users to perform are graphing, data mapping and visual spreadsheeting. These techniques are briefly described in the following sub-sections.

1. Graphing or Charting

Electronic spreadsheets allow us to visualize data in a lot of ways. One of the most important ways of visualizing data is by means of charting. Charting or graphing in spreadsheets is very simple. A user selects the data that has been graphed and then picks the appropriate commands from either the menu or the tool bar. Very often, user-friendly wizards guide the user through the process of setting up the graph. These wizards show a preview of the graph while providing dialog boxes for users to manipulate the settings of the graph.

A user can choose from several graph-types including column, bar, line, pie, XY, area, surface and bubble. It is also possible to combine some of these graph types. A user can pick the graph type that most effectively conveys the inherent nature of the data. For example, it is useful to visualize trends in time-series data by using a line graph. XY graphs are useful for depicting statistically related data. Pie graphs are useful for the

visualization of percentage-based allocations. Certain spreadsheets even offer the capability of drawing three-dimensional charts.

Typically, independent variable data to be graphed on the horizontal axis is specified as X-Series data. Dependent variable(s) data is specified as data series 1, 2, 3 and so on. The user can enter a chart title, sub-titles, axis titles and legends. Scales, major ticks, minor ticks, order of the data and the positioning of the titles can be specified too. Sophisticated spreadsheet software like Excel, allow the user to add trend lines and perform linear regression on the data by using just a simple right-click of the mouse on the chart data.

After a graph is setup, if the user changes any data in the worksheet, the instant recalculation feature of the spreadsheet recalculates all the formulas in the worksheet and updates the graph to reflect the new values. This capability allows users to conduct what-if analyses and study the impact of changes in the independent variables on a series of dependent variables to gain valuable insight into a situation. Such capabilities are enormously useful for sound decision making. Figure 7 contains a graph that has been created using the Microsoft Excel software. It depicts the stock price of Amazon Corp. over a three-month period. The daily high, low and the closing stock prices are plotted using the stock graph type.

Insert Figure 7 Here

2. Data Mapping

A simple Geographic Information System (GIS) or a spatial DSS can be implemented using a sophisticated spreadsheet like Microsoft Excel. Data mapping is as simple to perform as data charting. A user should enter labels for geographic regions like state names, county names or zip codes and the regional data values in a tabular form. Once the appropriate data is selected, simple clicks on the menu or the tool bar will lead to user-friendly dialog boxes guiding the user through the process of setting up the maps.

A gray-scale map of the United States depicting the population in the year 1990 is presented as an example in Figure 8.

Insert Figure 8 Here

3. Visual Spreadsheeting Using Influence Diagrams

An influence diagram provides a graphical presentation of the relationships in a model. Arrows are used to depict the influence of certain variables on other variables. The direction of the arrow indicates the direction of the influence. Typically, rectangles are used to represent independent or decision variables, circles are used to represent uncontrollable or intermediate variables, and ovals are used to represent dependent or outcome variables. An influence diagram can be drawn using the drawing toolbar that allows users the capability of super-imposing geometric shapes on a worksheet. Such diagrams allow the visualization of relationships between variables in a model that would normally be hidden inside formulas. An influence diagram for the car loan analysis previously discussed is presented in Figure 9.

Insert Figure 9 Here

4. Spreadsheets as DSS Generators

Computer-based information systems that combine models and data to enable users to solve semi-structured problems with extensive interactivity are typically referred to as Decision Support Systems (DSS). An integrated software package that provides capabilities for building specific DSS quickly, inexpensively and easily is known as a DSS generator. Two broad approaches can be identified in the evolution of DSS generators: one is that of special-purpose languages initially developed for mainframes and second is that of PC based integrated software systems.

Many commercial DSS generators evolved from financial planning systems with special-purpose languages modeled on natural language processing like Interactive Financial Planning System (IFPS) and Encore! Some others have roots in Database Management Systems (DBMS) like Nomad and Encore.

Spreadsheets are DSS generators in the sense that they allow decision situations to be modeled quickly, inexpensively and easily. Data can be manipulated by using models, what-if analyses can be performed and results can be visualized in tables, or by using graphs and maps. Users with Spreadsheets can solve semi-structured problems using all of the capabilities that have been described above.

I. Macros and Add-ins

To increase the productivity of a person using a spreadsheet, spreadsheet software allow automation of tasks through the use of macros. A user records a macro by turning the macro record feature on and then performing the steps that are to be recorded. Once a macro is recorded it can be played back to make the computer perform, as if the user is repeating, the tasks again. Sophisticated users can write their own macros in the spreadsheet software's macro programming language or edit the macros that have been recorded. Once the macros are recorded or written, they can be saved as templates or add-ins and then imported into other workbooks.

Visual Basic for Applications (VBA) is the common development language and environment found throughout Microsoft Office including Microsoft Excel and other VBA-enabled third-party applications. By providing a common development language and environment, Microsoft and other software vendors who license VBA for their applications enable developers to focus on the functionality of the applications instead of learning a new language for each application they incorporate into their solutions. Visual Basic for Applications is very similar to the Visual Basic programming language that developers use for developing applications for the Windows operating system and

to VBScript that people use for server-side scripting Active Server Pages and client-side scripting in Internet Explorer.

In addition to allowing developers to write macros in the macro program language, certain spreadsheets even allow modules written in other programming languages to be included as add-ins to enhance its capabilities. This capability of spreadsheet software to "add-in" modules has spawned an entire industry that develops software for use to enhance the functionality of spreadsheets. Table 1 lists some add-ins that are available for Microsoft Excel spreadsheet along with the functionality enhanced or provided.

Insert Table 1 Here

According to Fylstra et al. since its introduction in February 1991, the Microsoft Excel Solver has become the most widely distributed and almost surely the most widely used general-purpose optimization modeling system. A brief description of the capabilities of this popular and useful add-in along with an illustrative example is provided in the following sub-section. The next sub-section describes the use of the "Analysis ToolPak" statistical data analysis add-in.

1. Solver

Solver is an add-in that can be called from the menu of the spreadsheet program to determine the maximum or minimum value of one cell by changing another cell or a group of cells. For example, a user may ask Solver to figure out the advertising expenditures that generate the most profit. The cells that are selected must be related through formulas on the worksheet. According to its online documentation, Microsoft Excel Solver uses the Generalized Reduced Gradient nonlinear optimization code developed by Leon Lasdon, University of Texas at Austin, and Allan Waren, Cleveland State University. Linear and integer problems use the simplex method with bounds on the variables, and the branch-and-bound method, implemented by John Watson and Dan Fylstra, Frontline Systems, Inc.

Figure 10 illustrates an example of a product-mix decision that has been solved using Excel's Solver. The decision involves choosing the amounts to produce of two different products (cases of juice glasses, cases of wineglasses). The demand for juice glasses is forecasted at 800 units and the demand for wineglasses is forecasted at 1200 units. The production quantities should not exceed these forecasts (constraints). The net contribution from producing and selling each of these glasses is given. The decision-maker has taken into consideration two capacity limitations: the warehouse area where the products are stored and the machining time available. The decision-maker wants to figure out the quantities to produce of each product to maximize the profit generated without violating the capacity and the demand constraints. Figure 10 shows the Solver parameters where all this information is specified and also depicts the solution generated. Setting up the parameters is a simple exercise wherein a user clicks the relevant text box in the dialog box and selects the appropriate cells in the worksheet.

Insert Figure 10

2. Analysis ToolPak

Microsoft Excel provides a set of data analysis tools — called the Analysis ToolPak — that can be used to develop complex statistical analyses. A user provides the data and parameters for each analysis; the tool uses the appropriate statistical or engineering macro functions and then displays the results in an output table. Some tools generate charts in addition to output tables. Model specification is simple and is done by using wizards. Figure 11 illustrates the use of multiple regression analysis on bread sales data. A user is studying the relationship between the price of bread, amount spent on advertising and the sales.

Insert Figure 11

IV. Applications

Decision support including resource allocation, estimation, budgeting, and mathematical optimization account for the majority of spreadsheet applications. In addition, spreadsheets can be used for other business, government and personal applications. Common examples of applications developed in spreadsheet software include:

- financial investment analysis,
- pro-forma financial statement preparation,
- expense calculation and reporting,
- tax records and calculation,
- sales reporting and analysis,
- cash flow analysis, and
- cost-benefit analysis.

As mentioned in the introduction, millions of business analysts, managers and professionals around the world create spreadsheets each year. To get an idea of some of the current applications, a search was performed in the ABI/INFORM Global database covering periodicals published in 1998-1999 for the criteria "Spreadsheets and Applications" in the citation and abstract fields. Some representative applications resulting from the search are summarized below. Table 2 contains the citations to the works described below.

Insert Table 2 Here

Jackson and Staunton describe two applications of quadratic programming in finance. They show how, in the presence of inequality constraints, Excel's Solver can be used to find the optimal weights in both quadratic-programming applications. A direct analytic solution is implemented for generating the efficient frontier when there are no inequality constraints using the matrix functions in Excel. They show how Visual Basic for Applications can be used to program such tasks, confirming that techniques that were the preserve of dedicated software only a few years ago can now be easily replicated using Excel.

Friend and Ghobbar demonstrate that spreadsheets provide a low cost alternative to businesses interested in implementing materials requirements planning systems. The effective use of MRP is shown as a first step in the search for continuous improvement in maintenance and inventory control. The design for these MRP spreadsheets is based on data received from an airlines inventory system survey.

According to Vellani, crime analysis is the study of crime information to assist in decision-making and to enhance crime prevention and security programs. For the property manager, crime analysis is the logical examination of crimes that have penetrated a property's preventive plans and the application of revised policies and preventive measures that will hopefully lessen the recurrence of these crimes. Once the collection of relevant information is complete, it is time to conduct the crime analysis. The simplest method for data analysis is to use a spreadsheet software program, allowing for easy maintenance of the data and avoiding paper clutter.

Baker describes the application of spreadsheets to the assortment problem. Similar items of different sizes are required on a regular basis, but it is impractical to stock each of the different sizes. Demands for a size that is not stocked must be met by supplying the nearest acceptable size which is stocked, resulting in increased cost of trim wastage. A spreadsheet model is used to solve the problem in such a way that showing precedents on the spreadsheet results in the basis tree for the shortest or longest path solution being drawn without the need for special software.

Ruggiero argues that the versatility of a spreadsheet makes it a useful tool for market analysis. Spreadsheets allow complex calculations to be performed with market data and provide the ability to quickly test, chart and print the results of many "what if" scenarios. Using the programmability of the modern spreadsheet, it is possible to mechanize complex ideas and prototype these applications quickly. This paper

illustrates how to use a simple spreadsheet program (Microsoft's Excel) to test ideas and help build trading systems.

Erenguc and Erenguc describe an optimization-based audit-planning model, which was developed and implemented to formulate a biennial audit plan for the University of Florida. The model's objective is to formulate an audit plan that maximizes the total risk coverage while satisfying system constraints such as available auditor hours, minimum number of audits to be conducted, and the distribution of total audit effort between various types of audits. The objective function to be optimized is obtained by performing a risk analysis. The constrained optimization model is formulated as a spreadsheet application and solved using the SOLVER module of Microsoft's EXCEL.

According to Gyorki, when selecting motors, designers typically refer to manufacturers' data sheets. However, designers can enter manufacturer-supplied data into a spreadsheet application to calculate additional parameters, simulate motor performance graphically, and compare motor efficiency under varying conditions.

A paper by Hegji shows how a computer spreadsheet can be used to demonstrate the Dorfman and Steiner (1954) rule for optimal advertising. In their article, Dorfman and Steiner link optimal advertising to product price by a rule relating marginal advertising revenue to price elasticity of demand. The rule provides a simple guide for the overall utilization of firm resources.

Haussmann describes a situation wherein a spreadsheet macro language is programmed to control instruments and move data directly into a spreadsheet. With a VB add-on, ASCII command strings can be sent to control instruments through serial or other ports. The custom code allows the tester to pass I/O parameters and controls more than one instrument.

V. Academic Research

The premier source for information on Spreadsheet research is edited by Ray Panko of the University of Hawaii at URL http://panko.cba.hawaii.edu/ssr/. His site, called Spreadsheet Research, is a repository for research on spreadsheet development, testing, use, and technology. It has detailed abstracts for papers and some full papers. Most of the research focuses on spreadsheet errors because this is a very active research area and Panko's area of expertise.

What do we know about Spreadsheet errors? Panko (1998) summarizes results from more than 20 research studies including field audits, cell entry experiments, development experiments, and code inspection experiments. Uncorrected errors seem to occur in about 10% to 25% of spreadsheet applications.

Panko summarizes a number of studies. The following two seem the most important. Davies and Ikin examined 19 spreadsheets from 10 developers that were being used in 10 different firms. Four of the spreadsheets (21%) were found to have serious quantitative errors; 74% had structural problems; 68% had inadequate documentation. One error resulted in an inappropriate $7 million funds transfer between divisions. Cragg and King inspected 20 spreadsheets that were being used in 10 firms. Their audit found serious errors in 5 of the spreadsheets (25%). Developers and managers are generally overconfident about the accuracy of spreadsheets.

Some research has been done on the prevention and detection of spreadsheet errors. A web site at http://www.gre.ac.uk/~cd02/iirg/webspred.htm maintained by David Chadwick reports a pilot study that found error awareness, improved spreadsheet building methodologies/techniques, improved administration/control of spreadsheet developments, improved methods of costing errors and audit time helped prevent and detect spreadsheet errors.

Research indicates spreadsheets often have errors. Many consulting and accounting firms caution clients about end user developed spreadsheet applications. A marketing oriented web page at KPMG UK concludes "reports based on spreadsheets, databases and PC based systems may be unreliable and thus decisions based on them may be incorrect."

Most prescriptions for improving spreadsheet development and reducing errors have focused on modular construction, having an assumptions section for inputs, and using cell protection. There is some controversy about whether an assumption section reduces or increases errors. Clearly, linking data from an assumptions section makes the spreadsheet more complex and hence more subject to inadvertent errors. A related prescription by Panko is to "require developers to write out and critique all non-trivial formulas, doing calculations by hand, and then replicating these calculations with the formula typed into the spreadsheet." Finally, proper documentation seems to reduce the likelihood of errors in spreadsheets.

Interactive visible calculators have changed how managers, business analysts and many professionals perform their jobs. The quality and quantity of fact based analyses has increased in the past 2 decades. Spreadsheets are an accepted tool that all business students learn and many professionals need to master.

VI. Bibliography

1. Cragg, P. G., and, King, M. (1993). Spreadsheet Modeling Abuse: An Opportunity for OR? *Journal of the Operational Research Society*, 44(8), 743-752.
2. Davies, N., and, Ikin, C. (1987). Auditing Spreadsheets. Australian Account, 54-56.
3. Mattessich, R., "Budgeting Models and System Simulation," *The Accounting Review*, July 1961: 384-397.
4. Microsoft Excel Architecture - http://www.microsoft.com/Office/ORK/036/036.htm.

5. Fylstra, D., Lasdon, L., Watson, J., and, Waren, A., Design and Use of the Microsoft Excel Solver, *Interfaces*, 28(5), September, 1998.
6. Panko, R. What We Know About Spreadsheet Errors, *Journal of End User Computing's Special issue on Scaling Up End User Development*, 10(2), Spring 1998, pp. 15-21
7. Panko, R. Applying code inspection to spreadsheet testing, *Journal of Management Information Systems*, Fall 1999, 16(2), pp. 159-176.
8. Power, D. J., A Brief History of Spreadsheets, http://dssresources.com.
9. Russo, J. and J. Nafziger. "Software 'Look and Feel' Protection in the 1990's", copyright 1993, http://www.computerlaw.com/lookfeel.html.

Figure 1. A Screen Shot of the First Version of VisiCalc

Figure 2: Overview of Spreadsheet Concepts

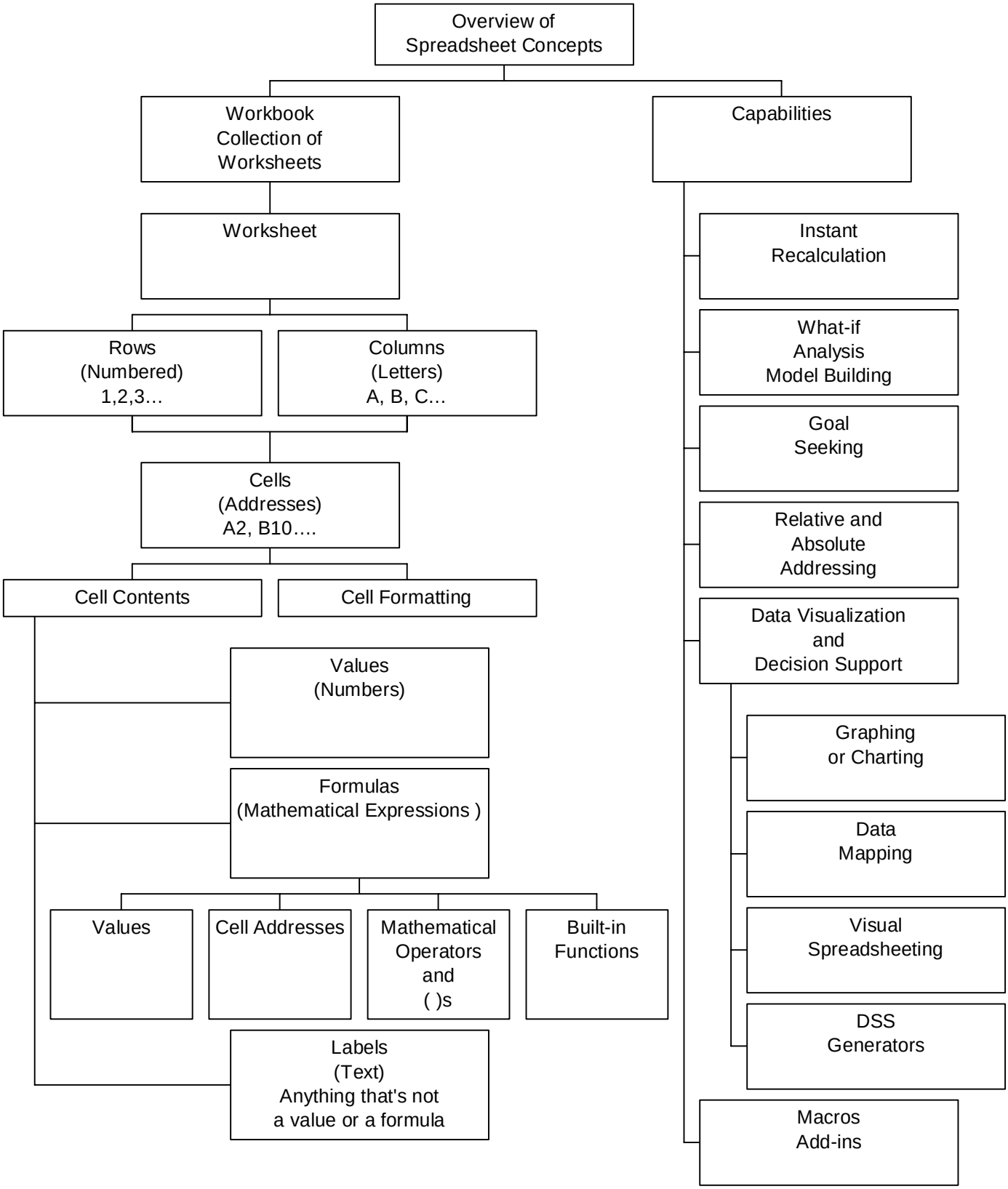


Figure 3: A Screen-Shot of Microsoft Excel

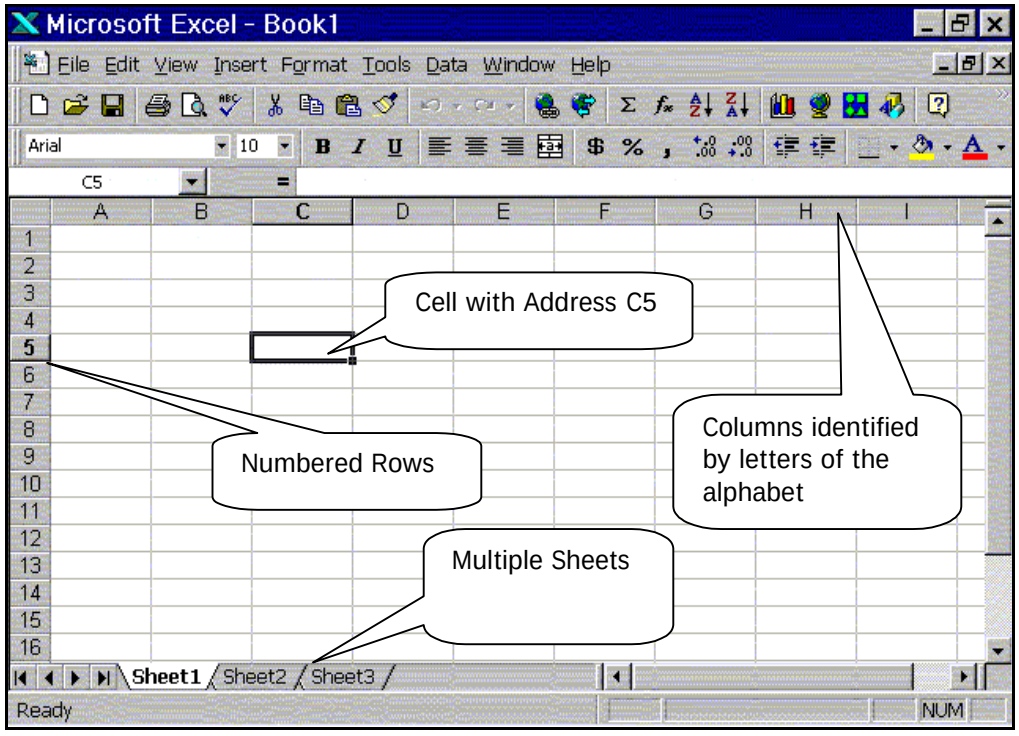


Figure 4: Using Excel to Analyze a Car Loan

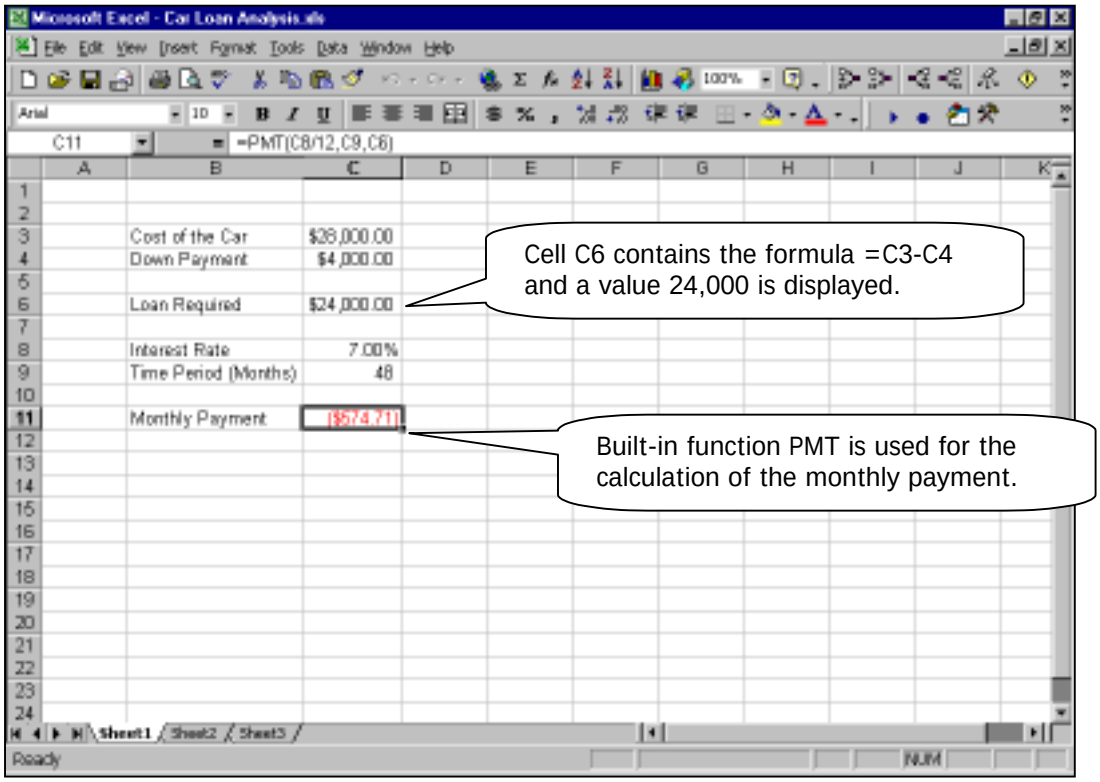


Figure 5: Car Loan Analysis Illustrating the Concept of Relative Addressing

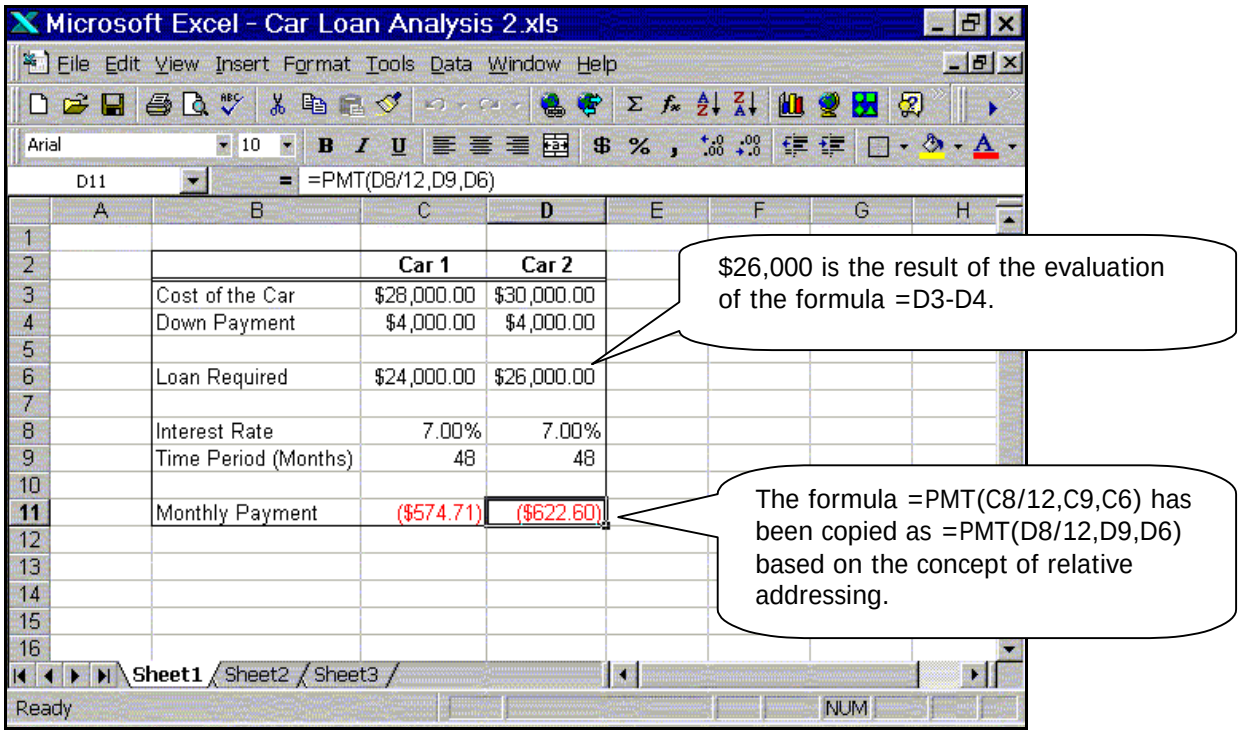


Figure 6: Relative and Absolute Addressing

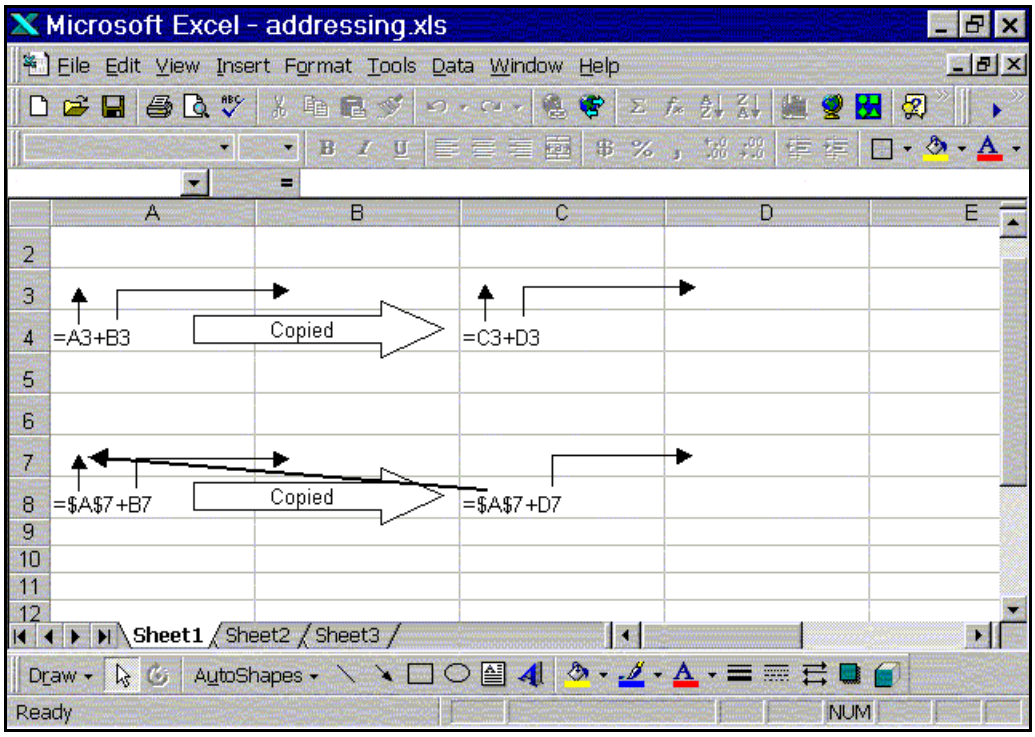


Figure 7: A Graph Created by Using Excel

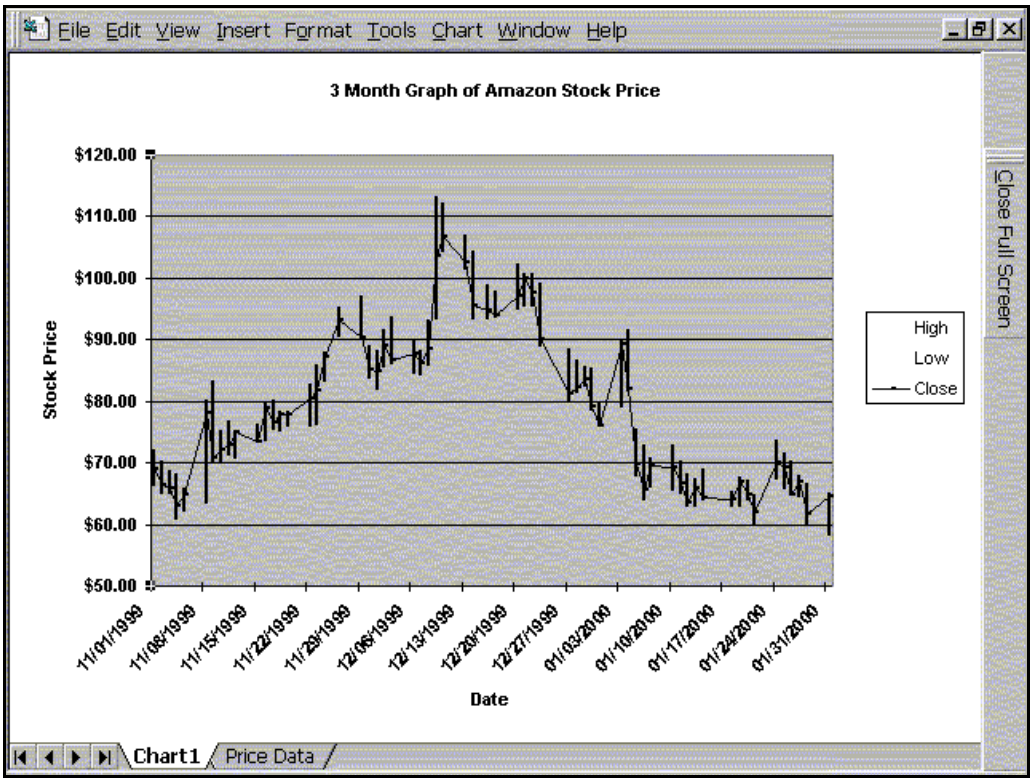


Figure 8: A Gray-Scale Map of the United States Depicting 1990 Population

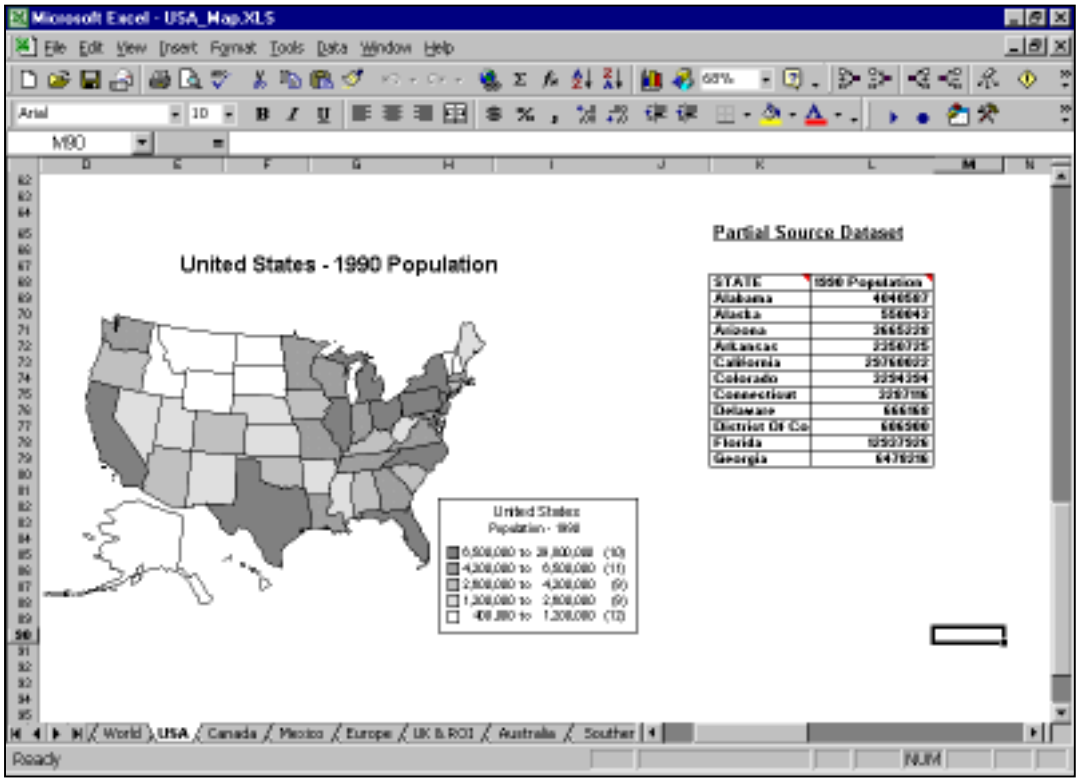


Figure 9: Influence Diagram of a Car Loan Analysis Model

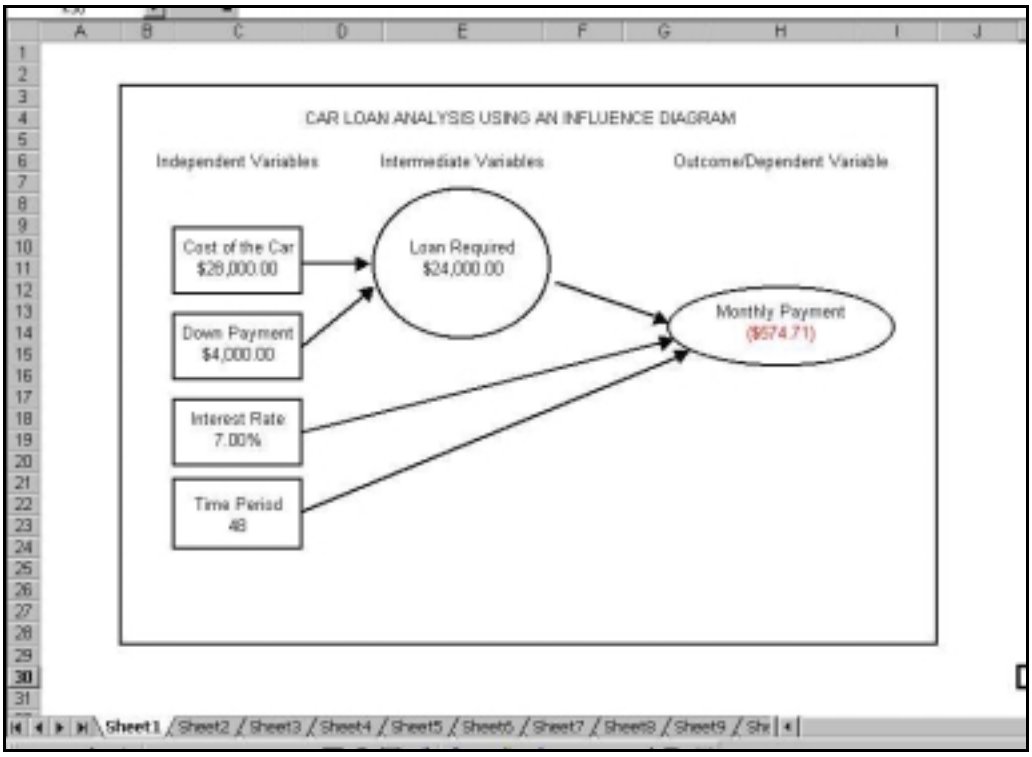


Figure 10: Solution to a Product-Mix Problem Generated by Excel's Solver

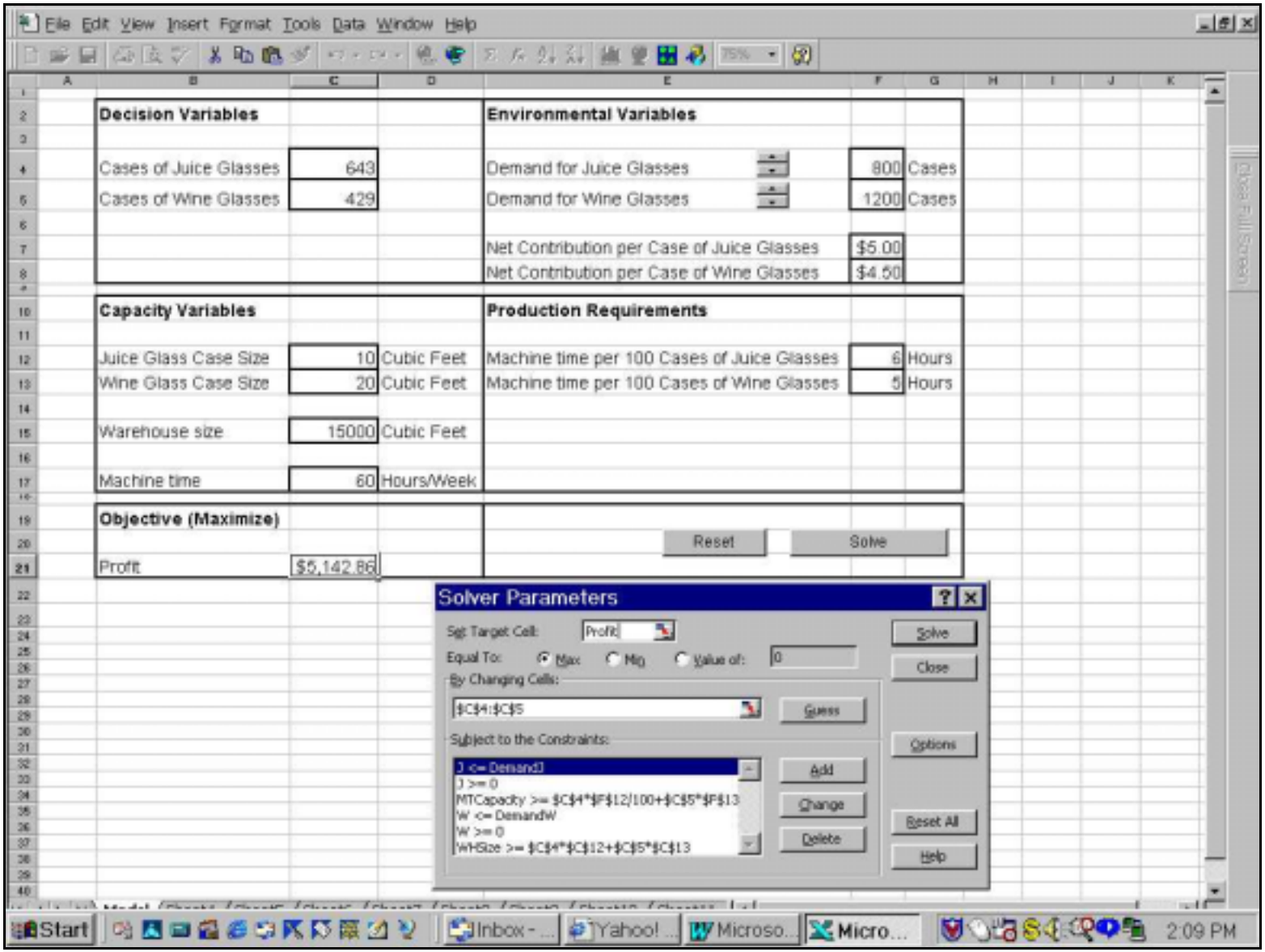
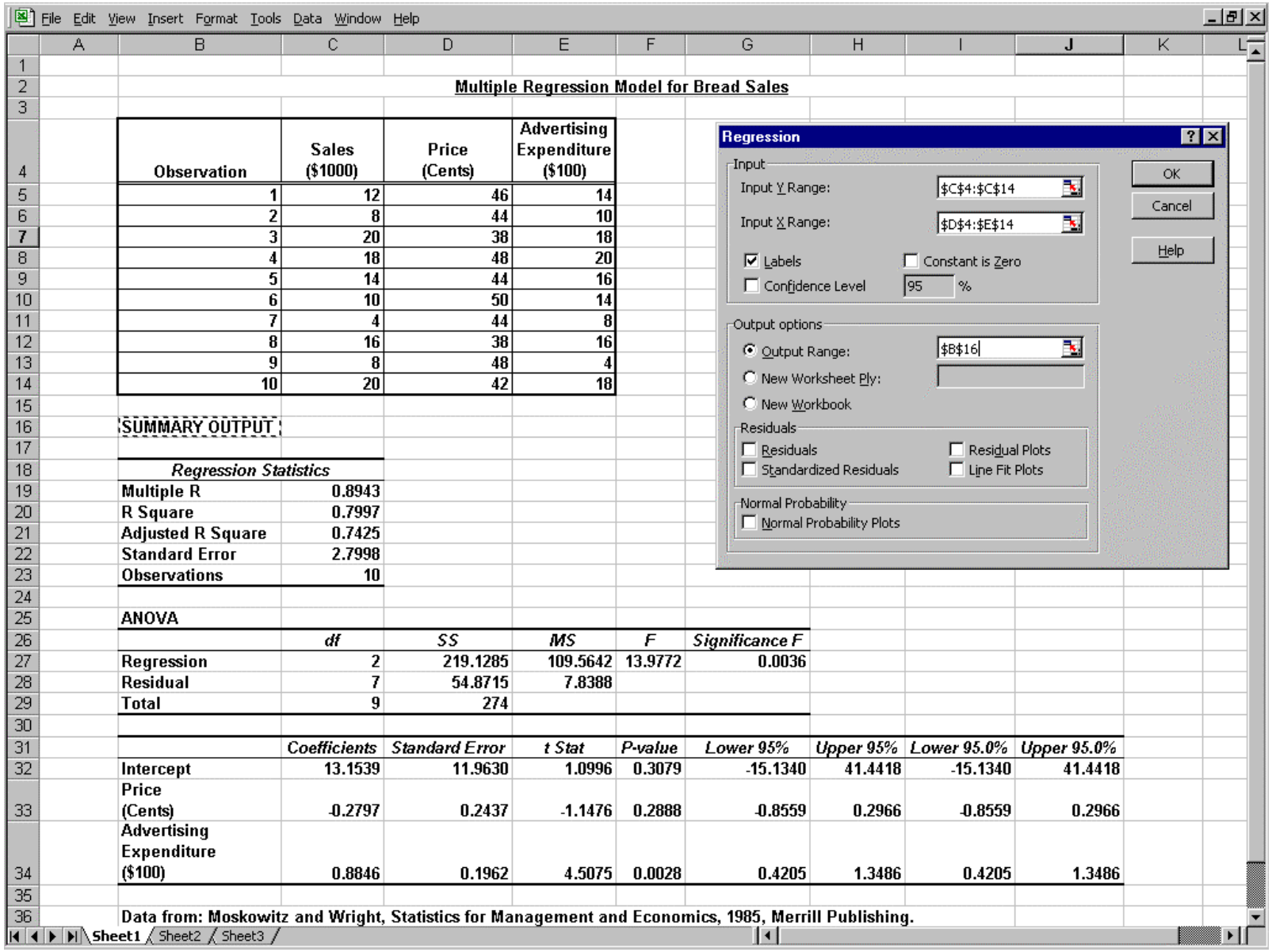


Figure 11: Multiple Regression Using Analysis ToolPak Add-in



Multiple Regression Model for Bread Sales

Observation	Sales ($1000)	Price (Cents)	Advertising Expenditure ($100)
1	12	46	14
2	8	44	10
3	20	38	18
4	18	48	20
5	14	44	16
6	10	50	14
7	4	44	8
8	16	38	16
9	8	48	4
10	20	42	18

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.8943
R Square	0.7997
Adjusted R Square	0.7425
Standard Error	2.7998
Observations	10

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	2	219.1285	109.5642	13.9772	0.0036
Residual	7	54.8715	7.8388		
Total	9	274			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	13.1539	11.9630	1.0996	0.3079	-15.1340	41.4418	-15.1340	41.4418
Price (Cents)	-0.2797	0.2437	-1.1476	0.2888	-0.8559	0.2966	-0.8559	0.2966
Advertising Expenditure ($100)	0.8846	0.1962	4.5075	0.0028	0.4205	1.3486	0.4205	1.3486

Data from: Moskowitz and Wright, Statistics for Management and Economics, 1985, Merrill Publishing.

Table 1: A Partial List of Add-ins Available for Microsoft Excel

Add-in	Description
Access Links Add-in	Allows you to use Microsoft Access forms and reports with Excel data tables.
Analysis ToolPak	Adds financial and engineering functions, and provides tools for performing statistical and engineering analysis.
Analysis ToolPak - VBA	Adds Visual Basic functions for the Analysis ToolPak.
AutoSave	Saves workbook files automatically.
Conditional Sum Wizard	Helps you create formulas to sum selected data in lists.
File Conversion Wizard	Converts several spreadsheet files to Excel format in one step.
Internet Assistant Wizard	Converts Excel tables and charts to HTML files.
Lookup Wizard	Finds values at the intersection of a row and column based on known values.
Microsoft Query	Retrieves data from external database files and tables using Query. (This add-in is used only when saving files in Microsoft Excel 97 & 5.0/95 (Windows) or Microsoft Excel 98 & 5.0/95 (Macintosh) format, or for backward compatibility for Visual Basic.)
ODBC	Adds worksheet and macro functions for retrieving data from external sources with Microsoft ODBC. (This add-in is included in Excel 97 (Windows) or Excel 98 (Macintosh) only for backwards compatibility; for programmatic data access, use DAO.)
Report Manager	Prints reports based on views and scenarios.
Solver Add-in	Calculates solutions to what-if scenarios based on adjustable cells, constraint cells, or cells that must be maximized or minimized.
Template Utilities	Contains utilities used by Excel templates.
Template Wizard with Data Tracking	Creates a template to export worksheet data to a database.
Update Add-in Links	Updates links in Excel version 4.0 add-ins to directly use Excel 97 (Windows) or Excel 98 (Macintosh) functionality.
Web Form Wizard	Helps you create an HTML form based on an Excel spreadsheet.

TABLE 2. Representative Works of Spreadsheet Applications

1.	Quadratic programming applications in finance using Excel The Journal of the Operational Research Society; Oxford; Dec 1999; M Jackson; M D Staunton;
2.	Extending visual basic for applications to MRP: Low budget spreadsheet alternatives in aircraft maintenance Production and Inventory Management Journal; Falls Church; Fourth Quarter 1999; Chris H Friend; Adel A Ghobbar;
3.	Crime stoppers Journal of Property Management; Chicago; Sep/Oct 1999; Karim H Vellani;
4.	A spreadsheet modeling approach to the assortment problem European Journal of Operational Research; Amsterdam; Apr 1, 1999; Baker, Barrie M;
5.	Trading in a spreadsheet Futures; Cedar Falls; Apr 1999; Murray A Ruggiero Jr;
6.	Optimization-based audit planning: A spreadsheet modeling approach Internal Auditing; Boston; Jul/Aug 1998; Erenguc, Nur S; Erenguc, S Selcuk;
7.	Simulating motor performance Machine Design; Cleveland; Feb 5, 1998; John R Gyorki;
8.	A spreadsheet application of Dorfman and Steiner's rule for optimal advertising Managerial and Decision Economics; Chichester; Feb 1998; Hegji, Charles E
9.	Custom control runs instruments from Excel Test & Measurement World; Boston; Jan 1998; Werner Haussmann;