Cost Analysis for a New Criminal Justice Complex

Jimish Gandhi
Greg Kehrier
Michael Olson
Goran Ristovski
Michael Widzinski

December 8, 2005

Abstract: This paper presents two Microsoft Excel applications developed for predicting the construction costs and annual operating expenses of a prospective criminal justice complex. The first uses a high level cost model, which estimates the construction and operating costs from a set of very basic inputs. The second uses a detailed cost model, which is much more rigorous than the high level model, but requires more extensive specifications. Thus, the high level model is intended to be used for a feasibility study, whereas the detailed model could be used to help develop a building design if the project is approved. Both applications serve as a tool for planning the design, construction, and operation of the prospective complex. The applications provide a default set of reasonable specifications, which are intended to be modified as more detailed and accurate information becomes available. When there is insufficient information to accurately determine some specifications, the applications use statistical distributions to account for the uncertainties. Then computer simulations are performed that calculate construction and operating costs. These statistical simulations enable users to perform sensitivity analyses and study "what if" scenarios. Both applications offer a high degree of flexibility so that users can customize them. A section on the site selection process is also included, along with a life cycle cost analysis for the project.

This work was proposed by the Institute for Public Policy and Social Research and the Michigan State University School of Criminal Justice to the Industrial Mathematics Program at Michigan State University. This project was advised by Richard Hensh of the Mathematics Department at MSU.

Table of Contents

Introduction

The purpose of this project was to conduct a feasibility study for a prospective criminal justice complex. This report presents two applications, a high level application and a detailed application, which were created to estimate the construction and operating costs of the prospective complex. Since these types of estimates are required to perform a cost-benefit analysis, these applications are very useful to determine whether or not the criminal justice complex should be built, and if so, to develop the building design.

A new construction project can be divided into four phases: the Feasibility Phase, Design Phase, Construction Phase, and Post-Construction Phase. A flowchart showing the four phases along with the main tasks performed during each phase is presented in Figure 1. The high level construction cost model will assist the owner in developing a preliminary cost estimate in the Feasibility Phase. By contrast, the detailed model will be useful when (if) the project reaches the Design Phase.

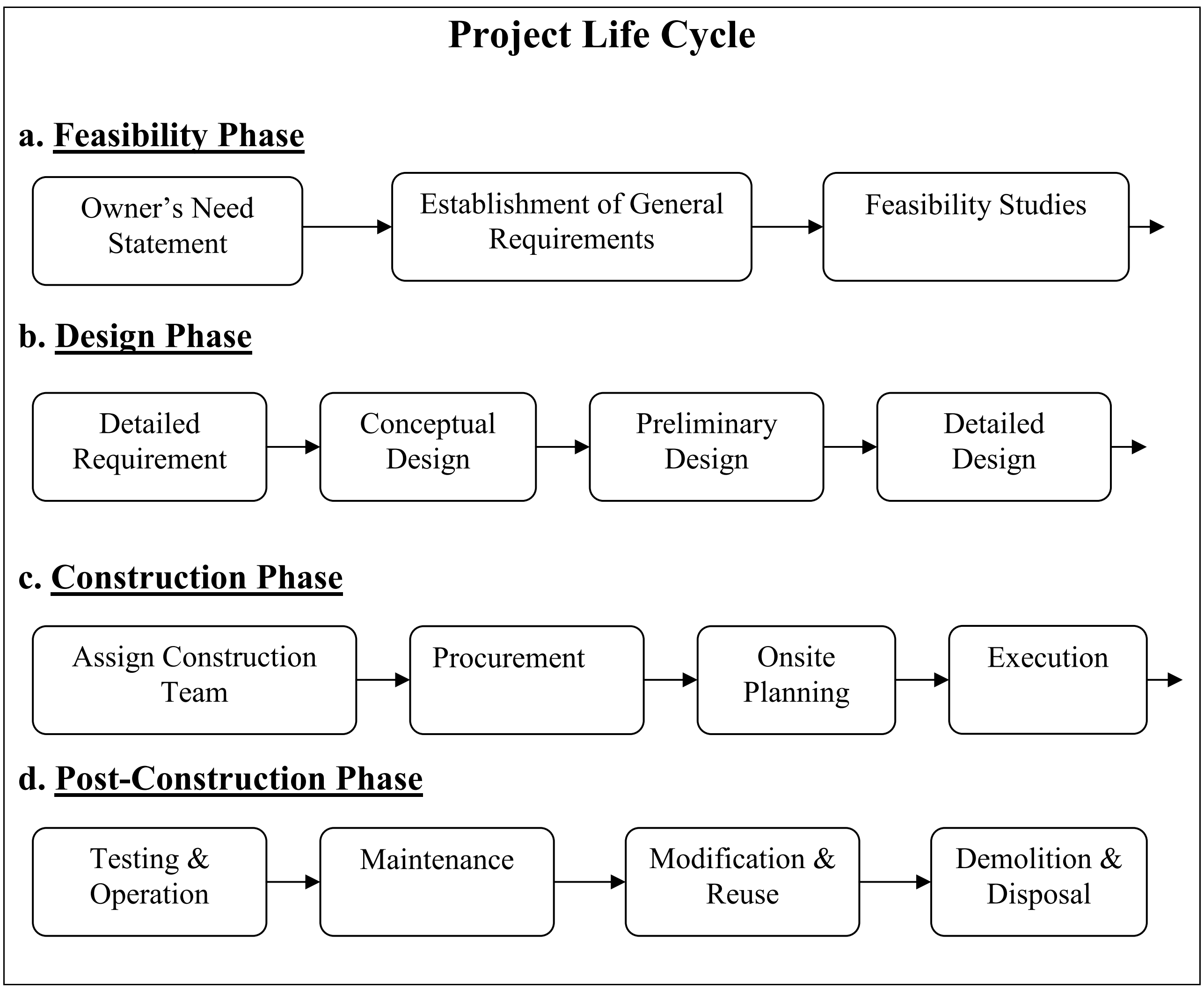


Figure 1. Flowchart showing the different phases of a Construction Project [16].

Both applications were created in the form of Microsoft Excel spreadsheets. The high level application estimates the construction and operating costs after the user specifies a very basic set of inputs. Conversely, the detailed application requires users to input more specific characteristics of both the physical building and the operations performed within it. These specifications are used to calculate both construction and operating costs using a more rigorous algorithm than the high level application. Therefore, the high level model is intended for use in the feasibility study, whereas the detailed model is particularly useful to develop the building design and operating strategy if the decision is made to build the complex.

When any required parameters are unknown, statistical distributions are used to model how the uncertainties affect the total construction and operating costs. This is accomplished in the high level application using custom-made Excel functions. In the detailed application, the statistical distributions are assigned using @RISK, a Microsoft Excel add-in created by the Palisade Corporation. Both applications perform thousands of computer simulations and calculate the average construction and operating costs of all the simulations. Including this statistical analysis enables users to determine the factors that most affect construction and operating costs, and to determine the levels of confidence associated with the estimates. The preliminary results from the high level application were used to perform a life cycle cost analysis that forecasts costs from groundbreaking through 20, 25, or 30 years of operation. To illustrate the application of these models, a cost analysis is presented for a new criminal justice complex for a major metropolitan area, viz. Detroit.

The current state of the Old Wayne County Jail, the Detroit Police Department headquarters and other criminal justice facilities underscores the need for a new centralized Criminal Justice campus [5]. The idea of a new centralized campus was initiated from the fact that building a new complex in Detroit city would decrease transportation and other operating costs. Moreover, it is believed that this new campus will decrease the overall annual operating cost significantly due to economies of scale. There is also a possibility that some of the space in this planned complex could be leased to other agencies, and this could serve as a source of additional revenue for Wayne County. The Detroit City Airport, the old Tiger Stadium and Michigan Central Depot are on the list of potential sites for the new complex.

The planning for this prospective project is still at a very early stage. There are no drawings for the building, and specifications are vague and in some cases not available. However, even at this early stage, cost estimates play a key role in planning efforts. The flexibility provided by the applications presented here allows preliminary building specifications to be developed.

Construction Cost Analysis

The construction cost model accounts for the battery limits complex, that is, all costs associated with buildings and equipment, but does not include costs associated with the construction site. For example, the model does not account for purchasing land, demolishing existing buildings, expanding roads, or upgrading utility capabilities. These are important factors that must be considered when the site is selected, but their consideration is not within this project's scope.

Detailed Application

The construction cost for the detailed application is divided into two sections, namely, the facility section and the equipment section. The facility section describes the costs for cell blocks, cafeterias, offices, hallways, bathrooms, court rooms, and so on. The equipment section describes the costs for items inside the facilities, such as the photocopiers and cubicles in the offices, the lockers and security systems in the cell blocks, and the dishwashers and grills in the kitchens. The analysis used for each of these two sections is described here, beginning with the equipment section.

The equipment cost section provides a simple means of calculating the total purchased equipment cost when the quantity and price of each item are known. The user inputs the quantity of each item to be purchased, along with its unit cost, and the total equipment cost is calculated. This method was selected because typically, all the necessary information to determine equipment costs can be readily determined, and there is no need for a more complicated algorithm. For example, suppose new desktop computers are to be purchased for all the offices in the complex. Then the quantity Q_E of computers is the basic specification, and the cost per computer P_E can be obtained easily from a vendor quote. Thus, the cost C_E for each line item in the equipment section is calculated by

$$C_E = Q_E \cdot P_E \,. \tag{1}$$

This computation is made for each equipment item, and the total equipment cost is added up.

Although a very simple model for the equipment cost is sufficient, the facility cost demands a much more robust algorithm. The facility cost accounts for the vast majority of the total construction cost, so this section is modeled more rigorously than the equipment section. Additionally, the facility cost is much more difficult to predict than the equipment cost, because a far greater level of specification detail is needed to accurately calculate the facility cost. Furthermore, many of the equipment and operating cost parameters are actually functions of the facility cost parameters. To name a few examples, the quantity of security cameras, the size of the cafeteria dishwasher, and the annual electricity cost will certainly all depend on the number of beds in the jail. Since the facility cost section is the most crucial part of the application, a more rigorous method was selected to model facility costs.

The facility cost algorithm uses three basic parameters to calculate the cost of each line item. The basic design specifications for each line item are the quantity and the size in square feet. For instance, suppose a certain type of single-occupancy jail cell is to be included in the complex. The user specifies that the complex will have (say) 500 of these cells, and each one will be 80 square feet. Then the only other parameter that is required to determine the cost of these cells is the price per square foot. Using these parameters, the total cost C_F for each individual facility is calculated by

$$C_F = Q_F \cdot S_F \cdot P_F \, , \tag{2}$$

where Q_F is the quantity, S_F is the size of each in square feet, and P_F is the price per square foot. For example, suppose these jail cells cost $100 per square foot. Then the total cost of all the cells would be 500 cells times 80 square feet each times $100 per square foot, which is $4 million. When all three of these parameters are known, the facility cost calculation is straightforward.

Unfortunately, the price per square foot is ordinarily difficult to specify, since its value depends significantly on the details of the facility in question. For example, the price per square foot of a jail cell likely depends on the design of the steel doors, locks, HVAC (Heating, Ventilation, & Air Conditioning), utility connections, and so on. The price per square foot would be larger if the cells had 12 inch thick concrete walls with high-security motion sensors on the doors than if they had 8 inch thick concrete walls and lower-grade security equipment. A great level of specification detail is required for a rigorous construction cost estimate, but most of the specifics are typically unknown during the early planning stage of a construction project. Thus, determining the price per square foot P_F, which is generally unknown for most facilities, is the most difficult aspect of the facility cost calculations.

To escape the dilemma that the cost per square foot is usually unknown, the uncertainty in P_F was quantified using statistical methods. In fact, this is precisely why equation (2) was selected to model facility costs. It is reasonably assumed that the quantity and size of each facility will be directly specified, and then all the unknown design details are simply transfigured into the price per square foot. Then, by researching the costs for similar existing facilities, the *most likely range of values* for P_F can be used, even though the *exact value* of P_F is unknown. From data on existing similar facilities, the most probable values of P_F can be inferred, and therefore the uncertainties are taken into account. The central idea of this type of analysis is to address the problem that most of the building specification details are unknown. Rather than merely estimating the facility costs with unknown certainty, the application calculates the *most likely* facility costs with quantifiable certainty. For example, the model results might indicate that with 90% certainty, the grand total construction cost will fall between (say, hypothetically) $200 and $650 million, and the most likely grand total construction cost is $400 million. The advantage of the statistical methods over rough estimation is obvious. Although the statistical methods complicate the facility cost algorithm, the added benefit is clearly worthwhile. A more detailed discussion of the statistical analysis is provided in Appendices C and D.

To implement probability distributions into the detailed model, the @RISK software add-in was used. Essentially, @RISK works by performing thousands of simulations of the criminal justice campus design. During each simulation, @RISK finds each of the uncertain parameters in the construction cost model, and randomly selects a value from a distribution (i.e. feasible range) of values. Then the results are added up to obtain the grand total construction cost. This process is repeated thousands of times, and the results are stored. But since each simulation is run with different values for the uncertain parameters, a different grand total construction cost is obtained each time. The unknown parameters are selected by favoring the most likely values, so when the simulations are complete, the range of most likely values of the grand total construction cost are known. The software creates a histogram of the grand total construction cost, which displays the mean grand total construction cost and the associated confidence intervals. Effectively, this simulation method is equivalent to watching a personal computer build the criminal justice complex ten thousand times. Then the user can see the range of values for the grand total cost, and the most likely value of the grand total cost.

Unfortunately, the detailed cost estimate approach described above has an obvious practical drawback. It is difficult to find data on existing facilities with sufficient detail for a range of values for P_F to be back-calculated. For instance, the cost per square foot of a cafeteria in a jail is not typically available from public sources. Since data this specific are not available, the default settings for P_F values in the detailed model are only rough estimates. Data for these types of values is typically the proprietary knowledge of architects or criminal justice facility consultants. Thus it is possible that the detailed application could be applied very effectively by these types of professionals, but it is not of immediate use for feasibility studies. There are simply too many unknowns. Therefore, the detailed application will be best applied during the Design Phase (see Figure 1).

However, even though the cost of the cafeteria in a jail is difficult to find, more general types of data for existing facilities are publicly available. For example, it is possible to find the number of beds, total size in square feet, and total construction cost for existing jails and criminal justice complexes, as well as qualitative facility characteristics. This notion of data availability suggests that the same statistical approach, if it is applied in less detail, would be useful for feasibility studies.

High Level Application

To estimate the construction costs without requiring extensive inputs, the construction cost C_C in the high level application is calculated by

$$C_C = B \cdot \frac{S}{B} \cdot P_F \,, \tag{3}$$

where B is the number of beds, S / B is the number of total square feet in the complex per bed, and P_F is the price per square foot of the complex. There is only one input parameter that is explicitly specified, namely, the total number of beds B. Statistical distributions are placed on S / B and P_F to form a range of values for each of these two parameters. This is clearly a much simpler algorithm than the detailed model. However, the general approach is the same, in that statistical distributions and simulations account for uncertainties. In this manner, the key advantage of the detailed model (modeling uncertainty) is retained, while the key disadvantage (too many unknowns) is removed. Therefore, the high level construction cost model generates results that are suitable for a feasibility study.

To calculate estimates without explicitly specifying S / B and P_F, the high level model includes a built-in set of data from some of the largest existing criminal justice complexes in the United States. For each existing facility, the total square feet S, total beds B, and total construction cost C_C were researched. From these data, values of S / B and P_F were calculated for each facility according to (3). Then the ranges of values were formed from the smallest and largest values of these two parameters. Since geographical location affects construction costs, the resulting P_F values were scaled for use in Detroit using cost-of-living data [13]. The year of construction was also taken into account by adjusting P_F for inflation. These calculations are all performed within the high level application on the Data worksheet, so that users can easily see how the researched data are used to calculate the range of values for S / B and P_F. The method used to adjust the costs to current Detroit values is also detailed in Appendix B.

Discounted Construction Costs

Both the high level and detailed applications include a discounted cash expense analysis for the construction cost. This enables the comparison of time-variant project expenses over three selected discounting periods. The equation for discounting the construction cost is

$$C_d = \frac{C_C}{N}\left(\frac{1-e^{-Nd}}{1-e^{-d}}\right), \tag{4}$$

where C_d is the annual discounted construction cost, N is the length of the discounting period in years, and d is the discount rate. By default, the model compares discounted costs for the three cases when the discounting period N is 20, 25, and 30 years. The default discount rate d is 3.2%, approximating the current rate of inflation. The user may adjust the discount rate and the three discounting periods as desired.

Operating Cost Analysis

As mentioned previously, two motivators for this prospective complex are to reduce operating costs in the justice system, and to obtain revenue from leasing office space and jail beds. The expected revenue is certainly an important consideration, but a revenue analysis this is not within the scope of this work. Similarly, no operating cost analysis of the current Wayne County criminal justice system is attempted. Instead, the goal is to estimate the costs of the prospective criminal justice complex. Certainly, the operating costs will depend on the financing methods used, as well as the construction site selected. The life cycle cost analysis (LCCA) and the site assessment guide included later in this report briefly describe these issues. However, this section focuses on the operating expenses that are independent of the financing methods and the site.

Detailed Application

The operating cost in the detailed application attempts to capture all relevant variable and fixed operating expenses for the prospective criminal justice complex. These include payroll (salaried and hourly employees), direct costs per prisoner, utility costs, maintenance costs, insurance, and so on. A default set of line items is included with the model, but the values are only rough estimates, and some values are left blank. The intention is that local knowledge can be leveraged to fill in the values, so that accurate results can be obtained. For example, the cost of electricity per kilowatt hour is presumably known, so users should specify this value. All the calculation algorithms are established, so after the input parameters are modified, the results are produced immediately.

High Level Application

Again, the idea is to estimate the operating cost using a small set of inputs. The high level application uses the number of beds B specified in the construction cost section to calculate an estimate for the annual operating cost C_O by

$$C_O = 365 \cdot B \cdot f \cdot C_P, \tag{5}$$

where f is the fraction of occupied beds and C_P is the total operating cost per prisoner per day. The percentage of capacity and cost per inmate per day are both modeled using statistical distributions. Data from existing facilities were used to establish the range of values for the cost per inmate per day [17]. Again, these values are adjusted for geographical location and inflation, and all the data and calculations are shown in the high level application on the Data worksheet. However, the percentage occupancy f is by default assumed to fluctuate between 75% and 105%. Users can adjust these values as desired.

How to Use the Applications

A complete user guide for both applications is provided in Appendix A, but an overview is presented in this section. It is assumed the user has basic familiarity with Excel. Specifically, the user is presumed to have the ability to change values in cells, navigate between worksheets, and understand basic Excel functions (e.g. SUM, AVERAGE, and so on). The user may refer to the Help topics included with Excel for descriptions of basic Excel features. The focus here is to describe how the unique features of these applications are used. For both applications, it is essential for users to read Appendix C to understand the statistical methods.

Both applications use macros and standard Excel add-ins. Macros must be enabled and the proper add-ins must be installed for the applications to function properly. The custom-made functions and routines in the high level application were created using VBA (Visual Basic for Applications), the programming language built into Excel. Therefore the high level model does not require any extra software to be procured. However, the detailed application uses @RISK, an Excel add-in which users will need to obtain from the Palisade Corporation [14].

Detailed Application

When the detailed application is opened for the first time, the user is directed to save the file with a different name. This ensures the default specifications are not permanently overwritten, and enables the user to start from scratch later on if desired. After creating a new file, the user is given the option to proceed with model creation using the "wizard" style graphical user interface (GUI), or to proceed without the GUI and navigate manually. Regardless of which choice is selected, the model creation process begins by developing the facility section of the construction cost model. The desired quantity and size of each type of facility (e.g. jail cells, courtrooms, cafeterias) are entered, along with the prices per square foot. When the model is used as intended, the following steps are taken. First, any unwanted line items are simply removed, any additional line items are added as desired, and the quantities of each facility are adjusted as necessary. Second, the user inspects each @RISK statistical distribution and makes reasonable modifications (see Appendix C). A similar approach is used to complete the equipment section of the construction cost model, as well as the parameters for the operating cost model. When all the specifications are complete, the @RISK simulations are started. When the simulations are completed (which normally takes about two minutes), the user consults the Summary worksheet to view the results.

High Level Application

The high level application opens displaying the construction cost section. Here, the user specifies the number of beds, and then selects from a pull-down menu the existing criminal justice complex that is most comparable to the intended Wayne County complex. The square footage per bed and cost per square foot of this comparable existing facility are displayed. Next, the statistical distributions for the remaining two input parameters can (and should) be modified. To assist the user with this, the DistCreate worksheet can be used, as described in Appendix C. (Again, it is crucial for users to read Appendix C to gain a basic understanding of the statistical methods used, so that their capabilities can be fully leveraged.) At this point, the construction cost specifications are complete, and the user proceeds to the operating cost section. Here, the user selects from another drop-down menu the most comparable facility (in the sense of cost per prisoner per day). This need not be the same facility selected in the construction cost section. Next, the percentage of holding capacity and the cost per inmate per day distributions are modified if desired. At this point, all specifications are complete, and the Simulate button is clicked. Results are generated in about 20 seconds.

Results

As mentioned previously, the default inputs for the detailed model are only rough estimates, or in some cases not available. Furthermore, the results of the high level model are of immediate use for feasibility studies. Therefore this section presents only the results from the high level model. However, the detailed model produces results of the same format as those presented here.

When the default distributions are used to run 10,000 simulations, the results are shown in Figures 2-A and 2-B, and Table 1. The histograms in Figures 2-A and 2-B show the frequency of each construction cost or operating cost, respectively. For example, the tallest bar in Figure 2-A shows that the construction cost was $300 million in about 670 of the 10,000 simulations. As another example, the third bar from the left in Figure 2-B shows that the operating cost was $55 million in about 320 of the 10,000 simulations. The "center of mass" of the shaded areas in these histograms represents the expected value, or *most likely value*, of the construction and operating costs, respectively. From the two figures, it is apparent that the most likely construction and operating costs are about $400 million and $100 million, respectively. These results are calculated precisely in Table 1, which also indicates the 90% confidence intervals. The confidence intervals indicate that, for example, 90% of the simulations resulted in construction costs between $155 million and $737 million. The discounted construction costs for periods of 20, 25, and 30 years are also calculated.

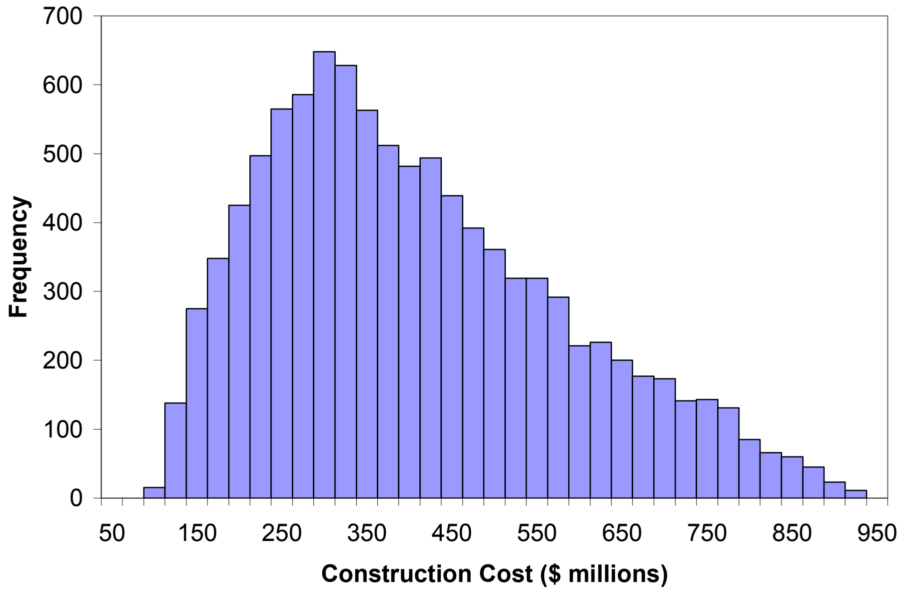


Figure 2-A. Construction cost results using default distributions.

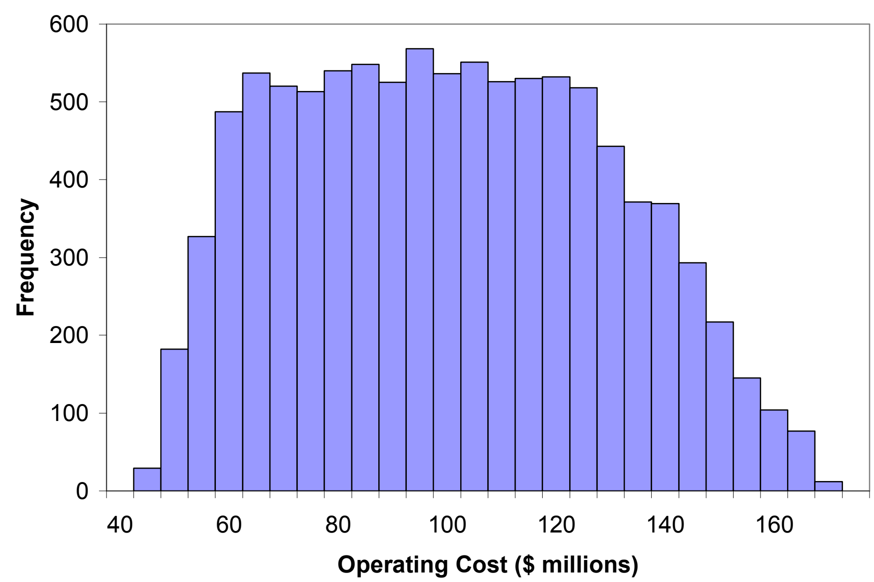


Figure 2-B. Operating cost results using default distributions.

Table 1. Simulation Summary (Default Distributions)

Construction Cost Results			
Minimum cost (90% confidence)	$160 million		
Expected construction cost	$410 million		
Maximum cost (90% confidence)	$732 million		
Operating Cost Results			
Minimum cost (90% confidence)	$54 million		
Expected operating cost	$101 million		
Maximum cost (90% confidence)	$147 million		
Discounted Construction Costs ($ millions)			
Discount period (years)	**20**	**25**	**30**
Minimum discounted cost per year	$120	$112	$105
Expected discounted cost per year	$308	$287	$268
Maximum discounted cost per year	$549	$512	$478

Discussion

The results presented in the previous section are obviously very broad. This is because they were generated using the default statistical distributions, which are designed to be modified by users to produce more meaningful results. Although a detailed explanation of this is given in Appendix C, the main point for this section is that the default distributions model the case when *nothing* is known about the prospective criminal justice complex. The intention is that users will compare the prospective complex to some existing facility, and modify the distributions accordingly. Doing so will yield much more informative results, as the following example demonstrates.

Suppose the user decides to examine the case when the prospective complex design will resemble that of the Los Angeles Twin Towers. That is, the user supposes the new complex and the L.A. facility will have similar offices and courtrooms, similar types of detention areas, comparable levels of technology, and so on. In this event, the values of S / F and P_F for the prospective complex are *most likely* close to values of the L.A. facility. Accordingly, the user modifies the default distributions so that the simulations are *biased* in favor of the L.A. values. The user tells the computer that within the feasible range, the most likely value is the one that corresponds to Los Angeles. Therefore the majority of the 10,000 simulations will be performed using values of S / F and P_F that are close to S / F and P_F for the L.A. facility. However, some simulations will still be performed using values in other areas of the feasible range, so that the uncertainty is still modeled. The effect is that the computer intelligently selects values from the ranges, accounting for the likelihood of each feasible value, so that the most likely values are selected most often. As expected, the results are much more specific.

For this example, the user modifies the default distributions so that the new distributions possess the following characteristics:

- Truncated logistic distributions;
- Minimum and maximum values equal to the least and greatest values from the data;
- Mean values equal to the L.A. facility values; and
- Uncertainty parameters of 0.05.

The user applies this type of distribution to the construction cost parameters S / F and P_F, and also to the operating cost parameter C_P. Again, the details are explained in Appendix C, but the idea is that the user decides to examine the case when the prospective complex is most comparable to the L.A. Twin Towers. In this case, the simulation outcome is shown in Figures 3-A and 3-B, and Table 2. When these results are compared to those of the default distributions from the previous section, the change is clearly a drastic improvement.

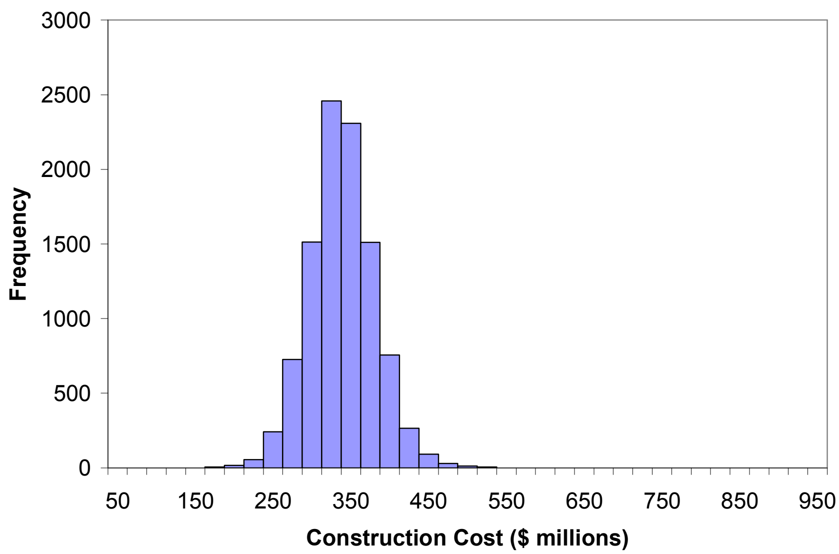


Figure 3-A. Modifying the default distributions produces much more specific results.

The example presented in the previous (Results) section and the example presented here demonstrate that if the user makes some simple decisions, very informative results can be obtained. Better yet, as more information becomes available about the prospective complex, the high level application can be easily expanded to accommodate more detailed calculations. Thus, it is possible to develop the high level application into a "medium level application". This can be done by simply adding more detailed calculations into the spreadsheet, and then updating the formulas for the totals under "RESULTS' on the HiLevModel worksheet. The macros are designed to allow for more rows of calculations to be inserted. As long as no existing contiguous ranges of cells are split up, the HiLevModel worksheet can be rearranged or expanded in any way. Then the simulation procedure can be used to determine the effect of new information on the results. Therefore the high level application could be used throughout the Feasibility Phase of the project (as shown in Figure 1) to develop cost estimates as more information becomes available. On the other hand, when (or if) the project reaches the Design Phase, the detailed application could give a fairly accurate estimate of the construction and operating costs. As mentioned previously, the detailed application could be used by an architect or a criminal justice facility consultant, so that proprietary knowledge of construction cost data could be leveraged.

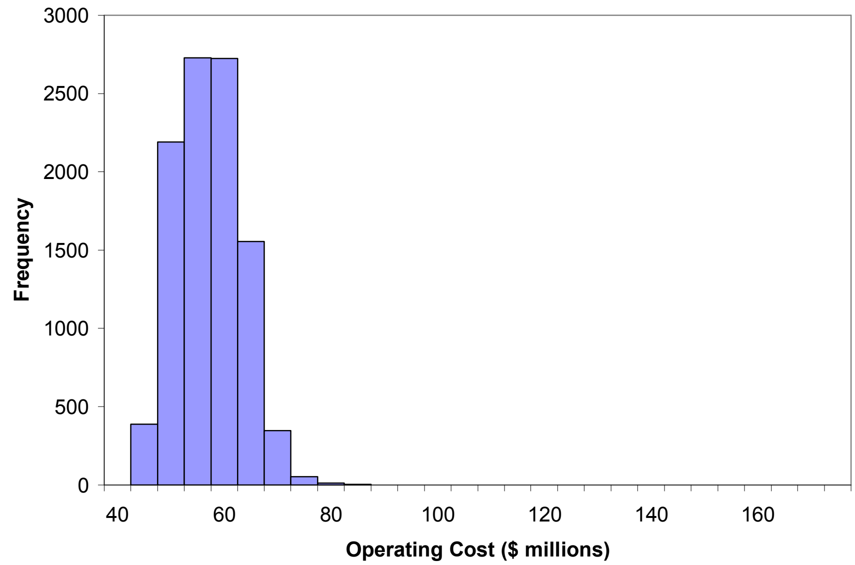


Figure 3-B. Modifying the default distributions shrinks the confidence intervals.

Table 2. Simulation Summary (Modified Distributions)

Construction Cost Results			
Minimum cost (90% confidence)	$259 million		
Expected construction cost	$338 million		
Maximum cost (90% confidence)	$395 million		
Operating Cost Results			
Minimum cost (90% confidence)	$45 million		
Expected operating cost	$57 million		
Maximum cost (90% confidence)	$64 million		
Discounted Construction Costs ($ millions)			
Discount period (years)	**20**	**25**	**30**
Minimum discounted cost per year	$194	$181	$169
Expected discounted cost per year	$254	$237	$221
Maximum discounted cost per year	$296	$276	$258

Site Selection Guide

In this section, factors affecting the site selection process are described. The three primary factors affecting the site selection process are size, location and cost [7]. The location of the jail also influences construction and annual operational costs. Site selection for a jail is often a political issue. Finding a location acceptable to the public is a major hindrance to many projects, resulting in delays, and in some cases, the termination of an otherwise well-planned effort. Some of the factors affecting the site-selection process are described in this section. All potential sites must be evaluated using the factors described below. A score ranging from 1 to 10 can be assigned to every sub-factor affecting the potential sites. Scores for all the sub-factors should be added to calculate the total score. The site with the highest total score should be selected.

Sub-factors Affecting Site Size

The size of the site is one of the primary factors influencing the site selection process. Some of the sub-factors affecting the site size can be briefly described as follows:

1. Building Area

A new jail is typically constructed to be four to six times larger than the old facility. However, the capacity of the jail does not increase four to six times. This is due to several factors besides the increase in bed capacity, including

- Increased per bed square footage requirements in housing areas;
- Increased provision of program and support areas; and
- Oversized support and program areas to accommodate future bed capacity expansion.

Also, to achieve interior efficiency, many jails have irregular shapes that create unusable spaces between portions of the building. If the size of the site is not adequate, a multi-story building can be developed. This might include a basement level for mechanical systems, computer rooms and storage. While constructing a multiple-level jail, the functioning and the security of the building should not be compromised. Also, for a multiple-level jail, it is critical to identify the functions that must be on the ground level.

2. Building Expansion

New jails should be designed in a way so that future capacity expansion is possible. Due to periodical changes in criminal justice philosophy, state law, and arrest rates, expansion planning must be part of any facility development process. Some of the future expansion options that should be considered while establishing the size of the required site are:

- Future bed capacity expansion;
- Future jail support/program service expansion;
- Future expansion of non-jail functions sharing the building or site;

- Law enforcement;
- Courts and court services;
- Clerk of courts and court services;
- Prosecuting attorney;
- Probation; and
- Other county offices.

It is common to provide twice the area of the jail for expansion.

3. Parking

The parking needs of a jail facility should be estimated using 350 to 400 square feet per car. One car per employee on the largest two shifts combined should be provided to allow adequate parking at shift change. An allowance of two parking spaces for the maximum number of visitors permitted at one time should be provided to allow enough parking for both the persons visiting and persons coming to visit during the next time period. Separate parking spaces should be provided for arresting officers, service providers (doctors, nurses, counselors, volunteers), official vehicles such as transport vans, and business and official visitors (lawyers, sales representatives, law enforcement officials). Pros and cons of providing multiple parking areas versus providing a single large parking area should be considered.

4. Outdoor Spaces

Most jails provide outdoor areas for their inmates, which include an exercise area, emergency refuge area, visiting area, garden area and general landscaping area. The exercise area should be walled, fenced, or unenclosed, depending on the security level of the facility. The emergency refuge area is a controlled outdoor area to which inmates can be temporarily evacuated in case of an emergency. The garden area is provided only if the facility grows some of its own food or has an inmate-training program. The landscaping area acts as a visual buffer between the building and the public.

5. Access to the Site

Internal roads for vehicles and pedestrian walks for the public occupy a significant portion of the land area. The principal forms of access for which area must be provided include the following:

- Arrestee delivery and transport in association with a secure vehicle sally port and a secure pedestrian entry;
- Food service delivery;
- Supply delivery;
- Staff;
- Public/official visitors;
- Inmates, staff or visitors with disabilities;
- Work release inmates, if they must enter and exit at a point different from most inmates for security reasons;
- Garbage removal;

- Emergencies, including medical and fire; and
- Officials from other agencies, such as law enforcement.

6. Support Elements

Additional space may be provided for the following, if applicable:

- Radio antennas;
- Gas tanks and pumps;
- Utilities (transformers, sewage treatment, etc.);
- Water retention areas;
- Garbage dumpsters; and
- Special law enforcement needs (vehicle service station, etc.).

The total site area should be determined by adding together all the elements. Thus,

Total Site Area Required = Building Area
+ building expansion area
+ parking areas
+ outdoor spaces
+ walk and drive access areas
+ support element areas.

Sub-factors Affecting Site Location

Location is a crucial factor to be considered while evaluating potential sites for a new facility. Some of the sub-factors affecting the site location can be briefly described as follows.

1. Linkages

The jail should be directly linked with the courts, sheriff's law enforcement offices, services like health care centers, educational resources, employee homes, etc. Additionally, proximity to emergency services such as the fire department and medical emergency facilities is desirable. The jail should be easily accessible to all staff working in the facility, as well as to visitors and others. The jail should also be accessible to the public transportation systems in the area.

2. Surroundings

It is important that the jail is located in a proper area. This will avoid conflict with different types of functions. Preferably the new facility should be located in a light industrial, government or commercial area. In general, it should not be close to a school, a residential area, a church, or a recreation area.

3. Views and Contact

Controlling views and contact between the inmates and the people of surrounding area is a very important consideration in site selection and design. Release of inmates in outdoor exercise areas involves a high risk. Inefficient planning and design of the site will increase the possibility of an inmate escape or contraband passage across fences. To help resolve this problem, use of high solid walls, protective screens over the exercise area, and careful placement of the exercise area can be effective. Also, positioning cell window sills well above floor levels, creating a heavily landscaped visual buffer, using tinted or reflective glass in windows, using inaccessible skylights or clerestory windows, and placing windows to look out onto controlled areas are some of the ways to avoid view conflicts from inmate occupied areas. Another contact related issue is that officers will bring arrestees to the jail daily, often with the arrestee in an agitated state. To prevent arrestee escape, it is common to provide a fully enclosed vehicle sally port.

4. Existing Jail Sites

Existing jail sites are often considered natural sites for new facilities because they are generally in good locations and already accepted by the public for jail use. However, there are some issues involved with this. Early demolition of an existing jail to make way for construction of a new facility may create major problems with transporting and boarding inmates elsewhere during the entire construction period, ensuring timely court appearances, having sufficient short-term holding capability for arrestees, and so on. Additionally, when the court houses and the jail are on the same site, planners must be careful not to limit growth options for adjacent court and criminal justice functions by putting the planning focus on the jail alone. In such cases, it is wise to consider doing a full justice system master plan to ensure that all needs are adequately accommodated prior to making site commitments. Finally, while considering potential sites for a new facility, historic status of buildings in the vicinity affected by the proposed project should be kept in mind.

5. Technical Requirements

Some of the technical factors that must be considered during the site selection process include the following:

- The type of soils on the site, which impact the height and cost of the building;
- Utilities available at the site or the cost of carrying them to the site;
- Topography (or slope), site drainage, and potential for flooding;
- Zoning, which could limit the height or square footage of the building or prevent its construction altogether; and
- Impact on traffic flow around the site.

Sub-factors affecting Site Costs

1. Acquisition Cost

This is the cost associated with the purchase of a proposed site. It may also include legal costs if condemnation proceedings are required to obtain certain parcels for the proposed site.

2. Demolition Cost

This is the cost incurred for demolishing existing buildings and removing materials from the site.

3. Utility Cost

These are the costs for bringing new utilities to a site, or relocating existing utilities that would otherwise interfere with proper building development. Gas, waste, storm sewer, telephone, electrical, and water lines should all be considered. Utility costs can vastly increase if new sewage treatment and water supply/treatment lines have to be provided at the site.

4. Site Preparation Cost

This may include significant grading of uneven areas, providing filling in low areas or areas with unstable soil conditions, and excavating rock to create basements or foundations.

5. Construction Impact Cost

These costs are imposed due to special conditions or restrictions on design or construction at the site. For example, if the site is in a flood plain, basement foundation walls have to be provided, which will incur additional costs.

6. Environmental Impact Study and Clean-up Cost

Generally, a consultant is hired to assess the impact of the proposed project on the surrounding environment. Evaluation considers environmental clean-up costs associated with cleaning soil contaminated by gas and oil products, chemical spills, etc.

7. Transportation Cost

These are expenses associated with providing the staff and the vehicles for transporting inmates between the new jail site and the courthouse or other detention, corrections, or service-providing facilities. A centralized complex would certainly reduce these costs.

8. Annual Energy Costs

Annual energy costs vary according to site characteristics. For example, presence of an aquifer that could be a source of chilled water would lessen air conditioning and heating costs.

Life Cycle Cost Analysis

To further quantify the operating costs of the prospective project, a life cycle cost analysis (LCCA) was performed. This is an economic method of project evaluation in which all costs arising from owning, operating, maintaining, and disposing of a project are considered. Since it is a powerful analytical process, the LCCA requires more information than analyses based only on first cost or short term considerations. It incorporates discounted cash flow, constant versus current dollars, and price escalation rates. Part of the LCCA is performed using a computational application called the BLCC (Building Life Cycle Cost) program, which is the program developed and recommended by the National Institute of Standards and Technology (NIST). This program could also be used to provide computational support for the analysis of the financial investment in the project.

The BLCC program is used to conduct economic analyses by evaluating the relative cost effectiveness of alternative building projects. Typically, BLCC software is used to evaluate alternative building projects that have higher initial costs but lower operating costs over the project life. The life cycle cost of two or more alternative projects are calculated and compared to determine what project has the lowest life cycle cost and is therefore more economical. The BLCC also calculates comparative economic measures for alternative projects, which include the following:

- Net savings;
- Savings-to-investment ratio;
- Adjusted internal rate of return; and
- Years to payback.

This program has been used to evaluate federal, state, and local government projects for both new and existing buildings.

Besides the BLCC, the DISCOUNT and ERATES programs were also considered in the process of conducting the Life Cycle Cost Analysis. DISCOUNT (Version 3.9) is normally used to compute the discount factors and related present values. Used along with BLCC, DISCOUNT is useful for solving life cycle cost analysis problems that do not require the reporting capabilities provided by the BLCC program. DISCOUNT could be used to perform the functions of standard discounting tables. It also has the capability to compute the following:

- Present values of future amounts;
- Future values of present amounts;
- Present and future values of periodic payments;
- Periodic payments corresponding to present and future amounts; and
- Corresponding discount factors.

ERATES (short for Electricity Rates – Version 1.0) calculates the electricity costs for a facility, using a wide range of electric utility rate schedules over a certain time period. It could also be utilized to estimate the future electricity cost of a facility.

Some of the most important parameters in the Life Cycle Cost Analysis include the real discount rate,

$$D = \frac{1+d}{1+i} - 1; \tag{6}$$

the present value for one time amounts,

$$SPV = FV(t) \cdot \frac{1}{(1+d)^t}; \tag{7}$$

the present value for annually recurring uniform amounts,

$$UPV = A_0 \cdot \sum_{i=1}^{n} \frac{1}{(1+d)^t}; \tag{8}$$

and the life cycle cost,

$$LCC = \sum_{i=1}^{n} \frac{C_t}{(1+d)^t}. \tag{9}$$

In these equations, A_0 is the uniform cost, $FV(t)$ is the future value at time t, C_t is the total initial cost of the project, and i is the inflation rate.

An example calculation for the Life Cycle Cost of the prospective complex was performed. In this example calculation, the parameters defined above had the following values.

- Construction Cost — $400 Million
- Operating Cost — $60 Million
- Remodeling Cost (Year 15) — $44 Million (estimate)
- Residual Value (end of Year 30) — $20 Million (estimate)
- Study Period — 30 Years
- Discount Rate — 4%
- Inflation Rate — 2.42%
- Real Discount Rate — 1.54%

The results of the life cycle cost analysis example are presented in Table 3.

Table 3. Life Cycle Cost				
Cost Items	**Base Cost**	**Year**	**Discount Factor**	**Present Value**
Initial Capital Investment	$400 Million	Base	None	$400 Million
Capital Replacement	$44 Million	15	SPV_{15} = 0.799	$35.15 Million
Residual Value	$20 Million	30	SPV_{30} =0.639	$12.78 Million
Annual Operating Costs	$60 Million	Annual	UPV_{30} =19.19	$1.15 Billion
Total Life Cycle Cost				**$1.60 Billion**

Conclusions

The following conclusions are supported by the results of this project.

- Applying a statistical analysis to the cost estimation algorithms enables the uncertainties in preliminary design work to be modeled and quantitatively understood.

- With 90% certainty, the construction costs for the prospective complex will be between $160 million and $732 million, and the most likely cost is $410 million. This wide range can be narrowed down significantly if the applications presented in this project are used as intended.

- With 90% certainty, the annual operating costs for the prospective complex will be between $54 million and $147 million, and the most likely cost is $101 million. Once again, this wide range can be made much more specific by using the models developed in this project.

- The high level model is best used to predict construction and operating costs for feasibility studies.

- The detailed model is intended to be applied during the Design Phase, and ideally with the guidance of an architect or criminal justice facility consultant.

References

[1] “Building Life Cycle Cost Program Version 4.3 – User’s Guide and Reference Manual, Petersen, Stephen, Building and Fire Research Laboratory, National Institute of Standards and Technology, Gaithersburg, Maryland, October 1995.

[2] Clayton County Georgia Web Site. Viewed November, 2005: http://www.co.clayton.ga.us/index.htm

[3] “Construction Cost Management Handbook,” Corrections Standards Authority, California Department of Corrections and Rehabilitation. Viewed August, 2005. Found online at http://nicic.org/Misc/URLShell.aspx?SRC=Catalog&REFF=http://nicic.org/Library/017087&ID=017087&TYPE=HTML&URL=http://www.bdcorr.ca.gov/cppd/more%20for%20less/more%20for%20less.htm

[4] “Corrections Planning Handbooks,” Board of Corrections, Guides for California’s Counties in Planning Adult and Juvenile Detention Facilities. Viewed October, 2005. Found online at http://www.bdcorr.ca.gov/publications/J-publication/Jail_Crowding_and_Planning_Resource/CorrPlanHdbk.pdf

[5] Gallagher John, October 2005. Article from Detroit Free Press, viewed online at http://www.freep.com/realestate/renews/justice5e_20051005.htm

[6] “Jail Design Review Handbook,” National Institute of Corrections, 2003. Viewed October, 2005. Found online at http://nicic.org/Library/018443

[7] “Jail Design Guide: A Resource for Small and Medium-Sized Jails”, National Institute of Corrections, 1998.

[8] “Life Cycle Costing Manual”, Fuller, Sieglinde; Petersen, Stephen, US Department of Commerce – Technology Administration, National Institute of Standards and Technology, NIST Handbook 135, 1995 Edition.

[9] Los Angeles County Sheriff’s Department Website. Viewed November, 2005: http://www.lasd.org/divisions/custody/twintowers/

[10] LTS Security Engineers, 1101 East Plano Parkway, Ste. H, Plano, TX 75074. Viewed November, 2005. http://www.lattatech.com/

[11] MacCluer, Charles R. Industrial Mathematics, Modeling in Industry, Science, and Government. New Jersey: Prentice Hall, 2000.

[12] Mecklenburg County North Carolina Web Site. Viewed November, 2005: http://www.charmeck.nc.us/Departments/MCSO/Divisions/Detention/Jail+Central.htm

[13] National Association of Realtors, Salary Calculator. Viewed November, 2005. Found online at: http://www.homefair.com/homefair/calc/salcalc.html

[14] Palisade’s web site. Viewed September, 2005: http://www.palisade.com/risk/

[15] “Planning Approval Process Handbook,” National Institute of Corrections. Viewed October, 2005. Found online at http://nicic.org/Misc/URLShell.aspx?SRC=Catalog&REFF=http://nicic.org/Library/013132&ID=013132&TYPE=PDF&URL=http://www.drc.state.oh.us/web/planapprove.pdf

[16] Syal, M.G. “Construction Project Management Course pack – Fall 2005”, Michigan State University.

[17] “The Corrections Yearbook, 1998”. Published by Criminal Justice Institute, Inc.

[18] University of Nevada at Las Vegas Web Site. Viewed November, 2005. Found online at: http://library.nevada.edu/arch/lasvegas/downtown01b.html#004

[19] Wayne County Sheriff Department's Web Site: http://www.waynecounty.com/sheriff/default.htm

Appendix A

User Guide

Detailed Model

Before you open the detailed model, some settings Excel must be properly adjusted. To do this, open Microsoft Excel, pull down the Tools menu, point to Macro, and select Security. Set the security level to Medium and click OK. Next, again pull down the Tools menu, but this time select Add-Ins. Select the check boxes for “Analysis ToolPak” and “Analysis ToolPak – VBA”. Click OK and follow the instructions on the screen (if any) to install these add-ins.

Next, open the file “Detailed_Model.xls”. The first window that appears on the screen is shown in Figure A-1 below. Click on Enable Macros.

Figure A-1. Click Enable Macros.

The detailed application will open with the dialog box shown in Figure A-2. This window asks you whether or not you prefer assistance to create a new model.

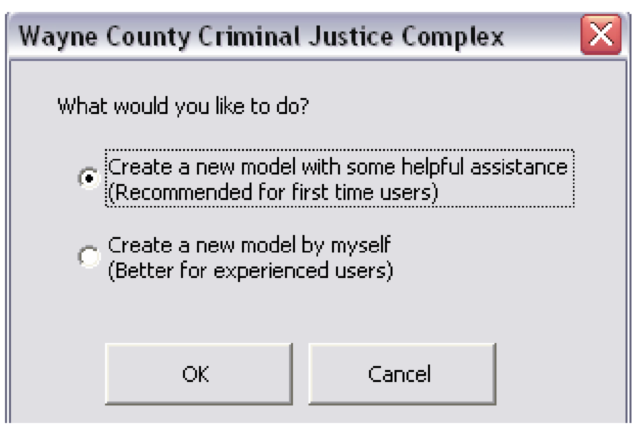


Figure A-2. Select an option and click OK.

If you select the second option, the user interface will close, and you can use the detailed application manually. If you are already familiar with the high level model, you may wish to select this option, since the syntax required in the detailed model is very similar.

If you select the first option, a "Wizard" style sequence will begin. The first dialog box in this sequence will direct you to save the file with a new name. Follow the directions on the screen to do this. Next, the Facility Cost form shown in Figure A-3 will appear on the screen.

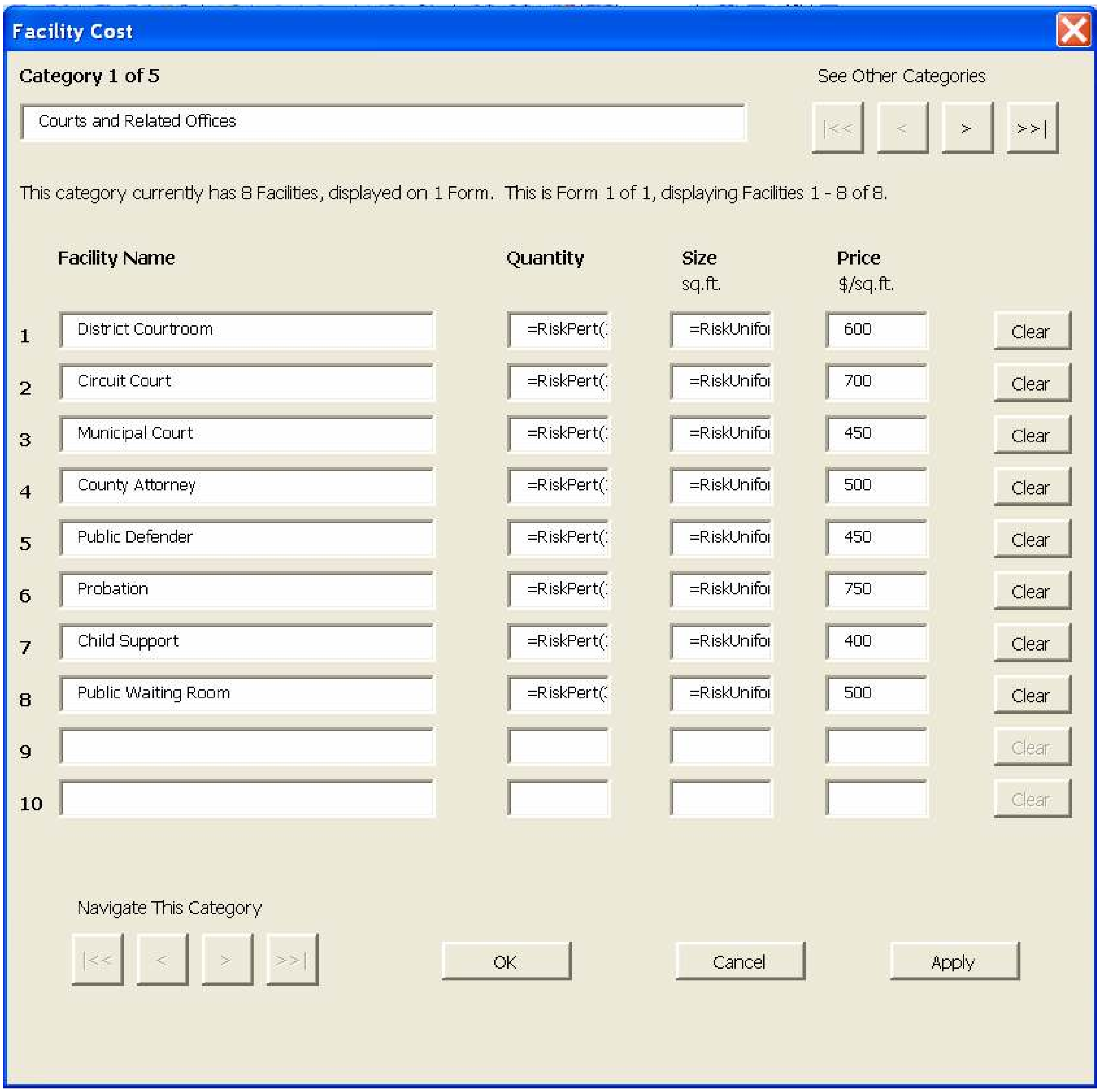


Figure A-3. Facility Cost Form.

The Facility Cost form is a convenient tool for looking at the distributions in many cells simultaneously. It is most useful the first time the model is modified, when many (if not all) the distributions will be changed.

The cell at the top of the form displays the title of the first category in the construction cost model. The "See Other Categories" buttons to the right of this cell allow you to navigate from one category to another. Clicking the extreme right button displays the last category, while the extreme left button displays the first category. The middle two buttons display the previous and the next category, respectively.

The middle portion of the form consists of four columns of cells. The first column (line item names) can be modified as desired. If a category has more than ten line items, the buttons labeled "Navigate This Category" can be used to switch to, say, line items 11 through 20. These buttons are arranged in the same format, and their functions are analogous.

The cells in the second through fourth columns allow the user to enter the quantity, size, and price for each facility. Cells that begin with "=Risk" contain @RISK distributions. You can modify as many of these cells as you like. When you are finished, you may click OK to load your changes into the spreadsheet and close the form. Alternatively, you may click Apply to load your changes into the sheet, but leave the form open to continue working. If you decide to start over from where you left off, you may click Cancel to close the form and reject your changes. Once you close the Facility Cost form, you may reopen it at any time by clicking the Open Wizard button, which appears on a custom-made toolbar. This button looks has a picture of the a Wizard's hat and the letters MW for Model Wizard, as shown in Figure A-4.

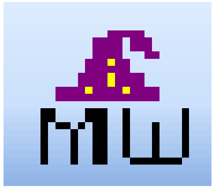


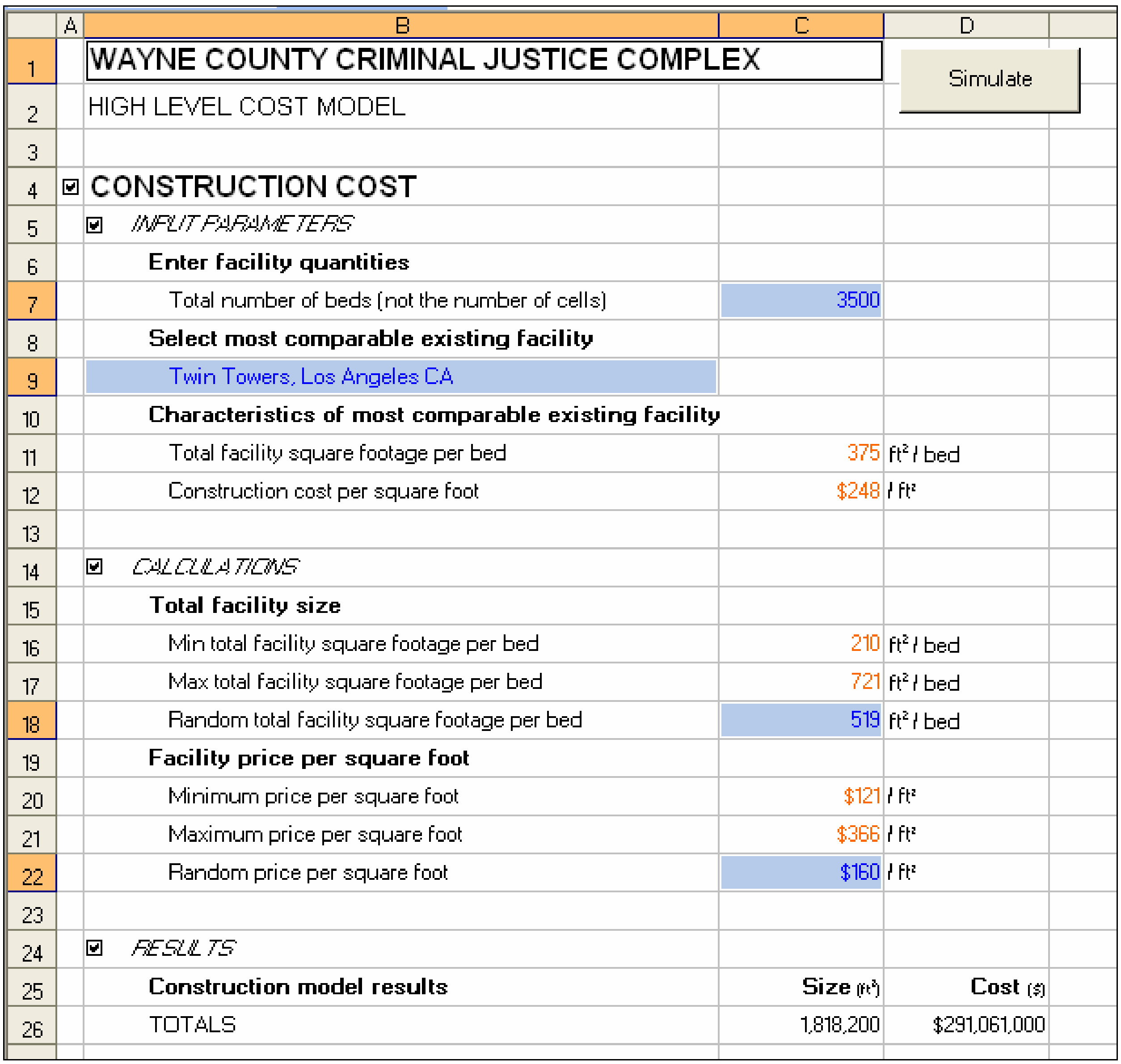
Figure A-4. Click the Open Wizard button to re-open the Facility Cost form at any time.

Once the facility cost section is filled in, the GUI's function is complete. The rest of the detailed model must be filled in manually. Proceed next to the equipment cost section on the Construction Cost worksheet, and use distributions in this section as desired. Finally, the operating cost worksheet may be filled out. When all the desired parameters have been entered, start the @RISK simulations. For assistance using @RISK, refer to the Help files provided with the @RISK software.

High Level Model

Before you open the high level model, refer to the beginning of this appendix to learn how to properly set the macro security. Ensure the macro security is set to Medium.

The high level model opens with the HiLevModel worksheet. This sheet consists of two main sections, the first of which is the CONSTRUCTION COST section displayed below in Figure A-5. Also notice there are check boxes on the sheet. You can click these boxes to show or hide the sections beneath them. You might also notice there are entries in columns K-M, in gray font. These are necessary to make the check boxes work, so they should not be modified.



	A	B	C	D
1		WAYNE COUNTY CRIMINAL JUSTICE COMPLEX		Simulate
2		HIGH LEVEL COST MODEL		
3				
4	☑	CONSTRUCTION COST		
5		☑ *INPUT PARAMETERS*		
6		**Enter facility quantities**		
7		Total number of beds (not the number of cells)	3500	
8		**Select most comparable existing facility**		
9		Twin Towers, Los Angeles CA		
10		**Characteristics of most comparable existing facility**		
11		Total facility square footage per bed	375	ft² / bed
12		Construction cost per square foot	$248	/ ft²
13				
14		☑ *CALCULATIONS*		
15		**Total facility size**		
16		Min total facility square footage per bed	210	ft² / bed
17		Max total facility square footage per bed	721	ft² / bed
18		Random total facility square footage per bed	519	ft² / bed
19		**Facility price per square foot**		
20		Minimum price per square foot	$121	/ ft²
21		Maximum price per square foot	$366	/ ft²
22		Random price per square foot	$160	/ ft²
23				
24		☑ *RESULTS*		
25		**Construction model results**	**Size** (ft²)	**Cost** ($)
26		TOTALS	1,818,200	$291,061,000

Figure A-5. The highlighted parameters and distributions should be modified as desired.

Cells in the high level model are color-coded as follows.

- Blue cells are input parameters or distributions, which the user should modify.
- Orange cells are values taken directly from the Data worksheet.
- Black cells are either headings or calculated parameters.
- Green cells (found on the Data worksheet) are values obtained by research.

The blue cells in the CONSTRUCTION COST section (highlighted in Figure A-5) are designed to be modified by the user. If you have not already done so, refer to Appendix C to understand distributions and to learn how to modify them.

First, enter the number of beds in cell C7.

Next, select the facility from the drop-down menu in cell B9 that you believe is the most comparable to the prospective complex. The characteristics of this facility are displayed in cells C11 and C12.

Next, modify the distributions in cells C18 and C22 as you desire. To help you visualize what kind of distribution to use, you can use the DistCreate sheet, shown in Figure A-6. The "DistCreate" worksheet allows the user to select a distribution type and then enter the parameters for that distribution. A graph displaying the shape of the distribution can also be seen on this worksheet. Directions are included on the worksheet itself.

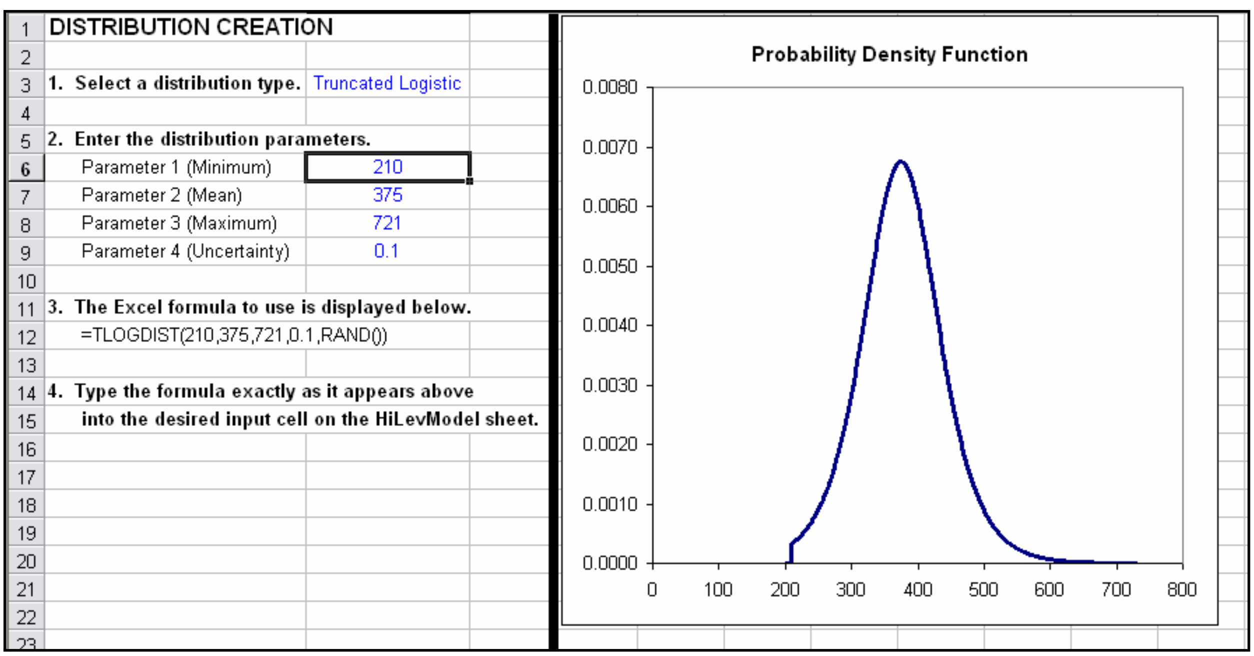


Figure A-6. Use the DistCreate worksheet to help you design your own distributions.

Return to the HiLevModel worksheet. After you have finished modifying the inputs in the CONSTRUCTION COST section, follow similar steps to fill in all the (blue) input parameters in the OPERATING COST section.

Move to the Options worksheet, shown below in Figure A-7. This sheet allows you to modify the simulation options and the discounting options. Modify these values however you with, but note that the number of simulations should be between 5000 and 20000.

	A	B	C
1		**WAYNE COUNTY CRIMINAL JUSTICE COMPLEX**	
2		USER OPTIONS	
3			
4		**Discounting Options**	
5		Year zero (i.e. year when construction will begin)	2007
6		Discrount rate	3.2%
7		Discount period 1 (years)	20
8		Discount period 2 (years)	25
9		Discount period 3 (years)	30
10			
11		**Simulation Options**	
12		Number of simulations to perform (should be 5000 - 20000)	10000
13		Confidence interval to report	90.0%
14			
15			

Figure A-7. Adjust the settings on the Options worksheet if you desire.

Return to the HiLevModel worksheet. Click the Simulate button at the top of the sheet. The simulations will begin, and you can watch the progress by looking at the status bar along the bottom of the screen. When the simulations are complete, four new worksheets will be created to process and display the results.

Appendix B

Data Sources

The data used in the high level model were obtained for the facilities listed in the first column in Table B-1. The year of construction listed in the fifth column is used to adjust the cost to current dollars. A multiplicative factor was used to adjust costs for geographical location, and this factor was assumed to be equal to the ratio of the costs of living [13].

The equation used to scale the data to current dollars in Detroit is

$$D = \frac{C}{S} e^{i(t_0 - t)} / L, \qquad \text{(B.1)}$$

where the variables are defined as follows:

D	Cost per square foot for a facility that would be built in Detroit;
C	The original total cost of the facility;
S	The size of the facility in square feet;
i	Inflation rate;
t_0	The year when construction on the prospective complex will begin;
t	The year when the facility was built.

Table B-1. Sources of Construction Cost Data

Data sources	S (ft^2)	B	S / B (ft^2/bed)	t	C ($ million)	L	D ($ / ft^2)
Twin Towers, Los Angeles CA	1,500,000	4,000	375	1996	3.7E+02	1.4250	248.13
Salt Lake Adult Detention Cmplx, (Salt Lake City, UT)	549,701	2,080	264	1999	1.2E+02	1.0270	276.86
St. Louis County Justice Center - Clayton, MO	500,000	1,232	406	1998	1.1E+02	0.9065	329.58
Riley County Law Enforcement Center - Manhattan, KS	58,870	120	491	2000	1.1E+01	1.2303	186.55
Randall County Detention Center - Canyon, TX	153,000	310	494	2002	2.0E+01	1.1494	130.12
Ramsey County Law Enforcement Center - St. Paul, MN	242,000	414	585	2004	4.5E+01	0.9246	221.38
Montgomery County Correctional Facility - Clarksburg, MD	305,000	1,100	277	2002	4.7E+01	0.9912	182.44
Johnson County Adult Detention Center - New Century, KS	120,000	264	455	1999	3.0E+01	1.2185	265.03
Garfield County Detention Center - Glennwood Springs, CO	55,000	200	275	2000	1.0E+01	0.7993	284.58
Dona Ana County Detention Center - Las Cruces, NM	112,000	400	280	1996	1.5E+01	1.2567	147.49
Cherokee County Public Safety Facility - Canton, GA	204,463	384	532	2002	2.7E+01	1.2797	121.10
Chatham County Detention Center - Savannah, GA	250,000	635	394	1992	2.8E+01	1.2488	144.94
Barron County Justice Center - Barron, WI	150,000	208	721	2003	2.6E+01	1.1146	176.75
Southeastern Correctional Facility - Charleston, MO	315,000	1,500	210	2002	7.3E+01	1.1136	244.21
Jefferson City Correctional Center - Jefferson City, MO	656,000	2,000	328	2004	1.1E+02	1.1031	171.13
Snohomish County Regional Justice Center - Everett, WA	251,993	640	394	2004	8.7E+01	1.0321	366.10
Mecklenburg Co. Jail (Mecklenburg Co., NC)	702,884	1,904	369	1997	1.4E+02	0.9330	298.82
Baltimore Central Intake Booking Facility (Baltimore, MD)	287,000	811	354	1995	4.9E+01	1.0570	234.72
Lexington-Fayette Detention Ctr. (Lexington, KT)	400,000	1,135	352	2000	6.2E+01	0.7750	250.21
Banke Justice Center, (Clayton Co., GA)	727,500	1,536	474	2000	9.1E+01	1.0700	146.25

Appendix C

User's Guide to Understanding Statistical Distributions

This appendix describes the importance of the user selecting appropriate probability distributions for use in the high level model. This is the most crucial factor that affects the results, so no user should overlook this section.

C.1 - Ranges of Values and Simulations

When the value of a design parameter is uncertain, the model assigns a *range of values* to that parameter. For the discussion in this appendix, the hypothetical example of an unknown parameter is the size of a courtroom. Since the exact size of each courtroom is unknown, the model assumes the size will be between the minimum and maximum values found from research, say, 4000 and 6000 ft^2. Then a size from that range is selected at random. The application performs similar random selections for each uncertain parameter. After all the uncertain design parameters have been randomly selected, the total size and cost of the complex are calculated. This entire process constitutes one *simulation*. The model performs thousands of simulations and keeps track of the results, which are used to generate the graphs shown earlier in the report.

C.2 - Probability and Probability Distributions

Suppose that researched data suggest it is rather unlikely that a practical courtroom could have a size outside the range of 4000 to 6000 ft^2. Accordingly, the model assumes that the *probability*, or likelihood, of a courtroom having a size outside this range is zero. The probability of a courtroom smaller than 4000 ft^2 is zero, and the probability of a courtroom larger than 6000 ft^2 is zero. By contrast, the probability of a courtroom being between 4000 and 6000 ft^2 is exactly one (1.0), because there is a 100% chance that the courtroom size will be somewhere between the practical bounds. Next, consider the likelihood *breakdown* within the feasible range. What are the most likely feasible sizes? Are large courtrooms more likely than small ones? If so, how much more likely are the large ones than the small ones? Alternatively, are the courtrooms no more likely to be large than small? To answer these important questions, there must be some way to define how the likelihood is spread across the range of feasible sizes. This definition is accomplished using a *probability distribution*.

Suppose, for example, that the user decides large courtrooms are more likely than small ones. In that event, the chance that courtroom sizes larger than 5000 ft^2 will be used is greater than 50%, or in other words, the probability that the courtrooms will be between 5000 and 6000 ft^2 is some number larger than 0.5. If a probability of (say) 0.8 were chosen for courtrooms between 5000 and 6000 ft^2, then the probability for courtrooms between 4000 and 5000 ft^2 is 0.2. This is because (by the laws of probability) the total of all the probabilities must always be 1.0. If a probability distribution with these characteristics were assigned, the model would perform 80% of the (say) 10,000 simulations using a courtroom size between 5000 and 6000 ft^2, and the remaining 20% of the simulations using a size between 4000 and 5000 ft^2. That way, the random size selections are *biased* intelligently, so that the most likely values are chosen more frequently than the least likely ones. Therefore, including probability distributions makes the simulation results much more realistic.

C.3 - Probability Density Functions

To incorporate a probability distribution into the high level model, a *probability density function* (PDF) is used so that the computer may precisely calculate how to bias its random selections. Probability density functions are defined using mathematical equations, but they can be easily understood by considering their graphs. The equations are developed in Appendix D, so only the graphical approach is discussed here. The graph of the PDF described in the last section is shown below in Figure C-1. This graph consists of two "rectangles", one between 4000 and 5000 ft^2 with a height of 0.0002, and the other between 5000 and 6000 ft^2 with height 0.0008. Recall that the probability distribution was defined with a 0.2 probability of courtroom sizes between 4000 and 5000 ft^2, and a 0.8 probability of courtrooms between 5000 and 6000 ft^2. Therefore, it is no coincidence that the *area* of the first rectangle is

$$(5000 - 4000) \cdot 0.0002 = 0.2\ ,$$

and the area of the second rectangle is

$$(6000 - 5000) \cdot 0.0008 = 0.8\ .$$

Thus, the total area enclosed between the graph and the horizontal axis is 1.0, as is the case with every probability density function.

Although it qualitatively describes the case when large courtrooms are more likely than small ones, this PDF has a key disadvantage. It is not *continuous*, but rather, its graph jumps from one "level of likelihood" to the next. Realistically, one expects a gradual transition rather than an

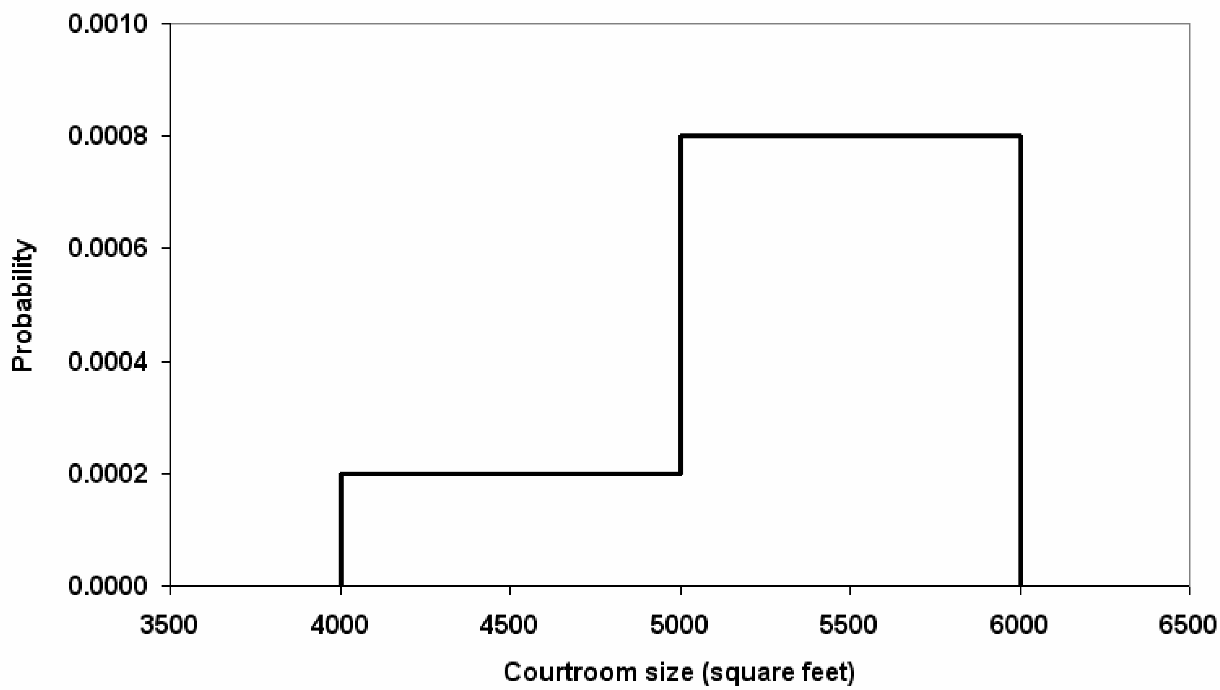


Figure C-1. If this PDF were used, the courtroom sizes would be between 5000 and 6000 ft^2 in about 80% of the simulations.

abrupt jump. So although the PDF from Figure C-1 was a useful example for discussion, only continuous PDFs were used in the application.

The remaining goal of this appendix section is to familiarize the user with the PDFs used in the high level model. After reading the remaining parts of this section, the user will understand how to select the most appropriate PDF for each situation, and how to modify the PDFs that are built into the model. In this way, the user can obtain much more meaningful results, rather than just very rough estimates.

C.4 – The Uniform Probability Density Function

The previous discussion focused on the case when large courtrooms are more likely than small ones. Now suppose instead that there is no specific information on the expected courtroom size. Large courtrooms are just as likely as small ones. In this case, there is no reason why the random size selections should be biased at all, so the *uniform probability density function* is most useful. An example graph of this PDF is shown in Figure C-2. From this graph, it is apparent that no value in the feasible range will be selected any more often than another. The user defines this PDF by merely specifying the minimum and maximum feasible values, e.g. 4000 and 6000 ft^2, respectively. The height of the rectangle (in this example, 0.0005) is automatically calculated so that the area under the graph is 1.0, as is required for all PDFs. The uniform PDF should be used to represent parameters for which very little specific information is known.

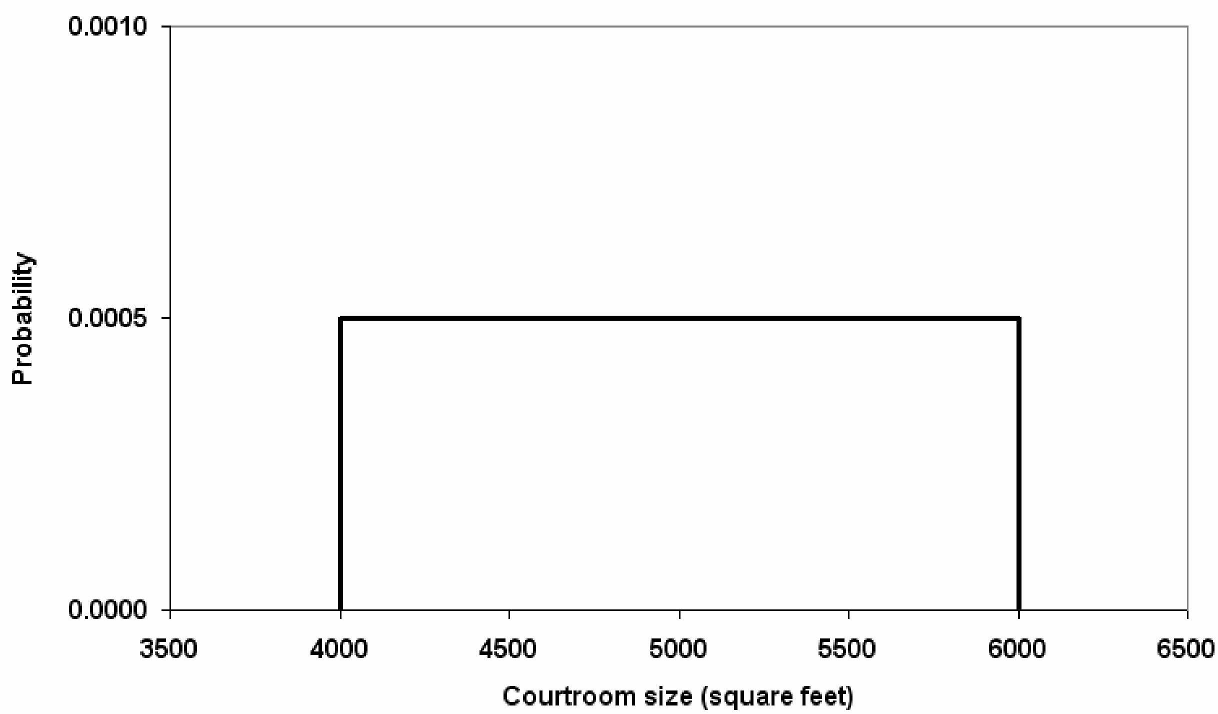


Figure C-2. The uniform PDF does not bias the random selections, so it is useful when very little information is known about the parameter in question.

C.5 – The Triangular and Truncated Logistic Probability Density Functions

Now suppose again that there is some reason to believe large courtrooms are more likely than small ones. Thus, the random size selection should be biased in favor of choosing a larger size. However, the PDF used to bias the selections should be continuous, unlike the PDF from Figure C-1. In such a case, a useful probability distribution is given by the *triangular probability density function*. A graph of this PDF is shown as the solid line in Figure C-3. To utilize this PDF, the user specifies the minimum, the *mode* (most often occurring value), and the maximum (which in this example are 4000, 5500, and 6000 ft^2, respectively). By examining Figure C-3, it is clear that 5500 ft^2 courtrooms will be selected most often, and the values in the middle of the feasible range will still be selected with a rather significant frequency. However, values near the ends will be sampled rarely. Therefore, this PDF is a good choice if values in the middle of the feasible range are more likely than values near the ends.

However, in some situations, the triangular distribution is not reasonable. For example, what if values near one end of the feasible range are the most likely, whereas values near the other end are the least likely? Or, what if one particular value in the feasible range is by far the most likely value? In these cases, the *truncated logistic probability density function*, shown as the dashed line in Figure C-3, is much more desirable. The example shown here heavily disfavors the less-likely smaller sizes, and biases the random selections toward the larger values. A particularly useful feature of the truncated logistic distribution is that it is more versatile than the triangular distribution. Whereas the triangular PDF has three adjustable parameters, the truncated logistic PDF has four. They are the minimum and maximum feasible values, the most likely value (denoted μ), and an uncertainty parameter (denoted β). In Figure C-3, these four

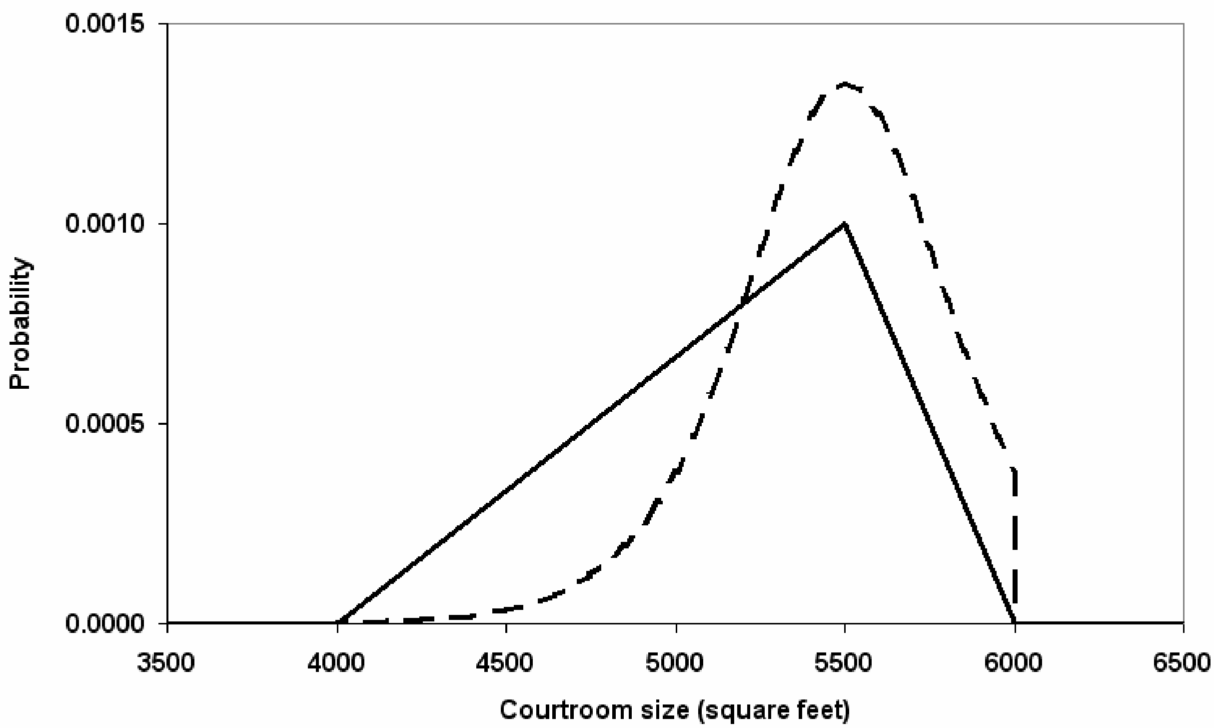


Figure C-3. The triangular PDF (solid line) and truncated logistic PDF are both useful for biasing random selections within the feasible range.

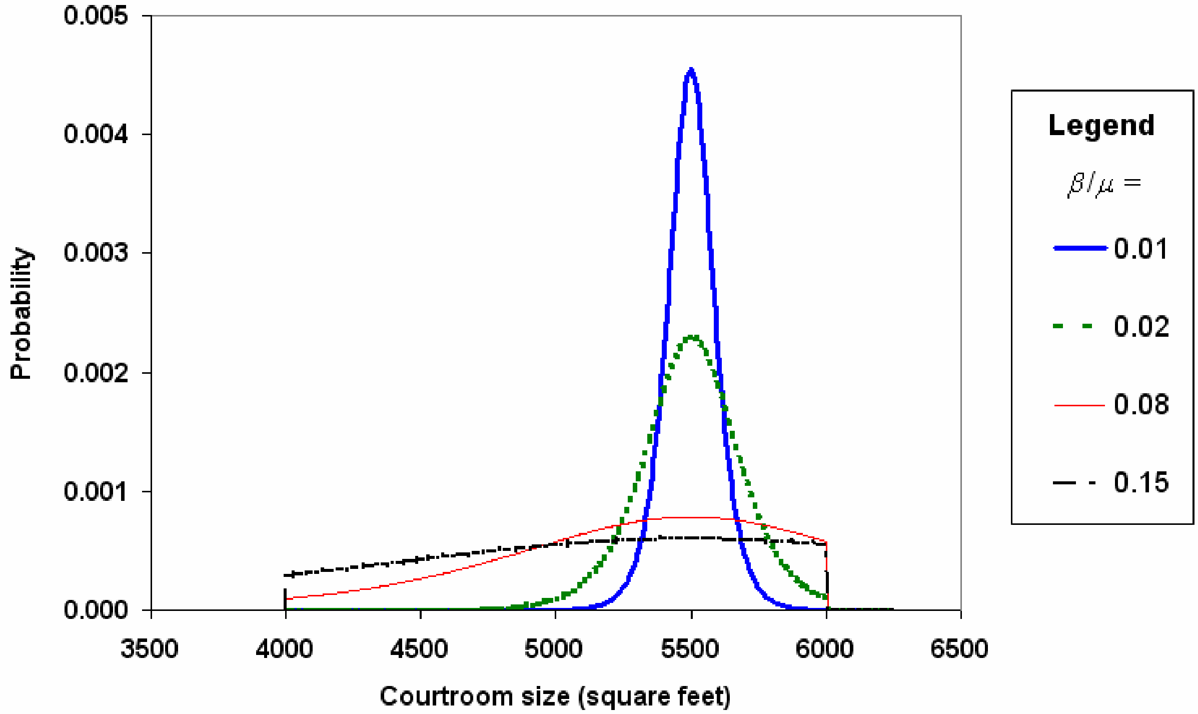


Figure C-4. The extent of biasing is determined by adjusting β/μ.

parameters are set at 4000, 6000, 5500, and 200 ft^2, respectively. Varying these four parameters enables the user to create a wide array of biasing levels. However, since the size of the uncertainty parameter β is meaningless unless it is compared to the mean value μ, it is more convenient to specify the uncertainty parameter using its ratio to the mean. Thus, the parameters used in the high level application for the truncated logistic distribution are the minimum and maximum values, the mean value μ, and the uncertainty ratio β/μ. The effect of adjusting the uncertainty ratio is illustrated in Figure C-4. Decreasing β/μ narrows the graph, and biases the random size selections so that the values close to μ are favored. For example, when $\beta/\mu = 0.01$ (solid, thick blue line), nearly all the randomly selected sizes will be very close to 5500 ft^2, and feasible sizes between 5200 and 5800 ft^2 are scarcely encountered. On the other hand, setting $\beta/\mu = 0.15$ (jagged black line) reduces the level of biasing and flattens the graph closer to the shape of the uniform distribution from Figure C-2. Clearly, the extent of biasing is rather sensitive to the value of β/μ. Therefore the truncated logistic distribution is very customizable, despite the fact that it is slightly more complicated than other distributions.

Both the triangular distribution and the truncated logistic distribution are very useful for biasing random parameter selections, but the most realistic results will be obtained when the user selects the most appropriate type of distribution for each uncertain parameter in the model. In the high level application, only the uniform, triangular, and truncated logistic distributions are used. The results are very sensitive to the type of distribution selected, so making a reasonable choice is very important. Fortunately, there are fairly obvious circumstances under which each distribution is best applied. These circumstances are summarized in the next section.

C.6 – Summary of Probability Density Functions Used in the Models

It cannot be overemphasized that the probability distributions are the most crucial aspect of the model. To obtain good results, it is absolutely imperative for the user to select a reasonable probability distribution. Otherwise, many of the simulations will be performed using values that are not very likely to occur, and the results will be extremely broad. However, if the user assigns reasonable probability distributions to each uncertain parameter, the simulations will be performed using the most likely values from each assigned range, and the results will be much more accurate. The remainder of this section explains when it is best to use each type of probability distribution.

It is a good idea for the user to have in mind the desired shape of the probability density graph before defining the distribution. With a rough idea of the situation in mind, the user should refer to Table C-1 to get a starting point. This table shows which type of distribution (uniform, triangular, or truncated logistic) to try, and also provides qualitative estimates for the distribution parameters. Next, the user may use the DistCreate worksheet in the high level application to graph the PDF they are considering. This worksheet allows the user to select any of the three distributions and experiment with the parameters to see their effect on the graph. After experimenting on the DistCreate sheet until the desired graph shape is obtained, the user may refer to Table C-2 to find the syntax of the desired Excel formula.

Table C-1. Rules of Thumb for Distribution Selection

Qualitative Description of Parameter Value Certainty	**Distribution Type**
Very little information is known, and there is no reason to believe any particular feasible value is most likely.	Uniform
Every feasible value is somewhat likely, even up to the end points, and (with little certainty) the most likely value is μ.	Truncated Logistic, large β/μ ($\approx 0.1 - 1.0$)
The end points are much less likely than the middle feasible values, and large values are more likely than small ones.	Triangular, mode closer to max than min
The end points are much less likely than the middle feasible values, and small values are more likely than large ones.	Triangular, mode closer to min than max
The most likely feasible value is closer to the large end of the feasible range, and large values are more likely than small.	Truncated Logistic, μ closer to max than min, $\beta / \mu \approx 0.05$
The most likely feasible value is closer to the small end of the feasible range, and small values are more likely than large.	Truncated Logistic, μ closer to min than max, $\beta / \mu \approx 0.05$
The most likely feasible value is almost certainly μ.	Truncated Logistic, small β/μ ($\approx 0.005 - 0.01$)

Table C-2. Excel Formula Syntax for High Level Model Distributions

Distribution	Syntax
Uniform	=UNIDIST([min],[max],RAND())
Triangular	=TRIDIST([min],[mode],[max],RAND())
Truncated Logistic	=TLOGDIST([min],[μ],[max],[β / μ],RAND())

Custom-made Excel functions have been created so that the distributions used in the high level model are easily implemented. The required syntax for these functions is shown in Table C-2. In each case, the parameters inside the brackets, e.g. [min] and [max], are replaced by numbers or cell references as desired. The last argument of each distribution formula is RAND(), which must be entered exactly as it appears in Table C-2 (with nothing inside the parentheses). The RAND() function generates a random number between 0 and 1, which the distribution function uses to generate a random number in the specified probability distribution. The distribution formula names are not case-sensitive, so the result will be the same whether the user types unidist, UNIDIST, or UniDist. These custom-made functions allow the distributions described in this section to be easily implemented.

If the output from any of these functions is the #VALUE! error, then either the wrong number of arguments was entered, or the arguments were entered in the wrong order. For example, the #VALUE! error will appear if the formula =TRIDIST(1000,4000) is typed, because the function requires three arguments and only two were entered. Similarly, the formula =TRIDIST(3000,4000,1000) generates the #VALUE! error because the arguments were entered in the wrong order. The mode must be between the min and max, and the max must be larger than the min.

Appendix D

Mathematical and Statistical Analysis

D.1 – Random Variables

Feasibility constraints require that the random variables take on values that are bounded. Therefore, for every random variable X in this project,

$$\text{prob}(a \le X \le b) = 1, \tag{D.1a}$$

and

$$\text{prob}(X < a) = \text{prob}(X > b) = 0, \tag{D.1b}$$

where a and b are the smallest and largest feasible values for X, respectively. Thus, all the cumulative distribution functions (CDFs) used to represent random variables have the form

$$F(x) = \begin{cases} 0, & x < a \\ \int_a^x f(z)dz, & a \le x \le b \\ 1, & x > b \end{cases} \tag{D.2}$$

where z is the dummy variable of integration, and $f(x)$ is a continuous probability density function (PDF). To ensure that F is continuous and strictly increasing on $[0,1]$,

$$\int_a^b f(x)dx = 1. \tag{D.3}$$

Then by the famous Central Limit Theorem of statistics,

$$X = F^{-1}(R), \tag{D.4}$$

where R is a random variable that is uniformly distributed between zero and one, inclusively.

This simple idea is employed to randomly generate random values from the probability distributions selected for use in the high level model. The three distributions selected were the uniform, triangular, and logistic distributions. The justification for use of the uniform distribution is obvious. The triangular distribution is useful to represent random variables that are much more likely to take on values on the interior of their feasible range than values near the endpoints. Certainly, many variables relevant to this project will be approximately normally distributed, but a problem would occur if the normal distribution were used. The distribution would need to be truncated; otherwise some values selected would not fall within the feasible range. Unfortunately, the normal distribution is difficult to analytically truncate, and thus the very similar logistic distribution was used so that an analytic truncation would be possible. For each of the three distributions in the high level model, the relevant equations are documented, and the VBA code used to program them into the model is included.

D.2 – Uniform Probability Distribution

The uniform probability distribution has a CDF given by (D.2) where $f(z) = 1/(b-a)$ if $z \in [a, b]$ and zero otherwise. So,

$$F(x) = \begin{cases} 0, & x < a \\ \dfrac{x-a}{b-a}, & a \le x \le b \\ 1, & x > b \end{cases}. \tag{D.5}$$

Applying (D.4), uniformly distributed random variables are generated by

$$X = R(b-a) + a . \tag{D.6}$$

The PDF, CDF (D.5), and inverse CDF (D.6) for the uniform distribution are implemented into the Excel model using custom-made functions called UniPDF, UniCDF, and UniDist, respectively. The VBA code for these functions is shown below.

```
Function UniPDF(Min, Max, Value)
If Value >= Min And Value <= Max Then
    UniPDF = 1 / (Max - Min)
Else
    UniPDF = 0
End If
End Function

Function UniCDF(Min, Max, Value)
If Value < Min Then
    UniCDF = 0
ElseIf Value >= Min And Value <= Max Then
    UniCDF = (Value - Min) / (Max - Min)
Else
    UniCDF = 1
End If
End Function

Function UniDist(Min, Max, Value)
If Value >= 0 And Value <= 1 Then
    UniDist = (Max - Min) * Value + Min
End If
End Function
```

As mentioned in Table C-2, the `Value` variable in the `UniDist` function is intended to be the RAND() command, which generates a random number from a uniform distribution on $[0,1]$. It is apparent from the code that any number in $[0,1]$ may be used for `Value`, but this function returns the #VALUE! error if `Value` $\notin [0,1]$ or if the arguments are otherwise improperly entered. Also, it should be noted that the function names are not case-sensitive.

D.3 – Triangular Probability Distribution

The CDF for the triangular distribution with mode *m* is given by equation (D.2) where

$$f(z) = \begin{cases} \dfrac{2}{b-a} \cdot \dfrac{z-a}{m-a} \ , \ a \le z \le m \\ \\ \dfrac{2}{b-a} \cdot \dfrac{b-z}{b-m} \ , \ m < z \le b \end{cases} \tag{D.7}$$

if $z \in [a, b]$ and zero otherwise. Thus, the CDF is

$$F(x) = \begin{cases} 0, \quad x < a \\ \dfrac{(x-a)^2}{(b-a)(m-a)}, \quad a \le x \le m \\ 1 - \dfrac{(b-x)^2}{(b-a)(b-m)}, \quad m < x \le b \\ 1, \quad x > b \end{cases}. \tag{D.8}$$

Applying (D.4), random values are selected from the triangular distribution using

$$X = \begin{cases} a + \sqrt{R(b-a)(m-a)}\ , \quad R \le (m-a)/(b-a) \\ c - \sqrt{(1-R)(b-a)(b-m)}\ , \quad R > (m-a)/(b-a) \end{cases}. \tag{D.9}$$

The VBA code for (D.7) – (D.9) is shown below.

```
Function TriPDF(Min, Mode, Max, Value)
If Value <= Min Then
    TriPDF = 0
ElseIf Value <= Mode Then
    TriPDF = 2 / (Max - Min) * (Value - Min) / (Mode - Min)
ElseIf Value <= Max Then
    TriPDF = 2 / (Max - Min) * (Max - Value) / (Max - Mode)
Else
    TriPDF = 0
End If
End Function

Function TriCDF(Min, Mode, Max, Value)
If Value < Min Then
    TriCDF = 0
ElseIf Value <= Mode Then
    TriCDF = (Value - Min) ^ 2 / ((Max - Min) * (Mode - Min))
ElseIf Value <= Max Then
    TriCDF = 1 - (Max - Value) ^ 2 / ((Max - Min) * (Max - Mode))
Else
    TriCDF = 1
End If
```

```
End Function

Function TriDist(Min, Mode, Max, Value)
If Value >= 0 And Value <= (Mode - Min) / (Max - Min) Then
    TriDist = Min + (Value * (Max - Min) * (Mode - Min)) ^ (1 / 2)
ElseIf Value <= 1 Then
    TriDist = Max - ((1 - Value) * (Max - Min) * (Max - Mode)) ^ (1 / 2)
End If
End Function
```

Again, the `Value` variable in the `TriDist` function is intended to be the RAND() command, but any number in $[0,1]$ may be used without a #VALUE! error.

D.4 – Truncated Logistic Probability Distribution

The (non-truncated) logistic distribution [ref] has the CDF

$$F(x)=\frac{1}{1+\exp(-(x-\mu)/\beta)}$$

and the PDF

$$F'(x)=f(x)=\frac{\exp(-(x-\mu)/\beta)}{\beta\cdot\left[1+\exp(-(x-\mu)/\beta)\right]^2}.$$

To make this distribution useful, it must be truncated so that (D.1a) and (D.1b) are satisfied. Thus, the truncated logistic probability density function $f_t(x)$ is defined so that

$$\int_{-\infty}^{a} f_t(x)dx = \int_{b}^{\infty} f_t(x)dx = 0. \qquad \text{(D.10)}$$

Since (D.10) strikes out the area under the curve of $f(x)$ to the left and right of a and b, $f_t(x)$ must be obtained by dividing $f(x)$ by the appropriate normalization constant η to satisfy (D.3). The normalization constant required is

$$\eta = 1-\int_{-\infty}^{a} f(x)dx-\int_{b}^{\infty} f(x)dx\,, \qquad \text{(D.11)}$$

so

$$f_t(x)=\frac{f(x)}{\eta}=\frac{f(x)}{1-\int_{-\infty}^{a} f(x)dx-\int_{b}^{\infty} f(x)dx}. \qquad \text{(D.11)}$$

Now the only remaining tasks are to calculate η, then integrate $f_t(x)$ to find $F_t(x)=\int_{-\infty}^{x} f_t(z)dz$, and finally find $F_t^{-1}(x)$. So first,

$$
\begin{aligned}
\eta &= 1-\int_{-\infty}^{a}\frac{\exp(-(x-\mu)/\beta)}{\beta\cdot[1+\exp(-(x-\mu)/\beta)]^2}dx-\int_{b}^{\infty}\frac{\exp(-(x-\mu)/\beta)}{\beta\cdot[1+\exp(-(x-\mu)/\beta)]^2}dx \\
&= 1-\left[\frac{1}{1+\exp(-(x-\mu)/\beta)}\right]_{-\infty}^{a}-\left[\frac{1}{1+\exp(-(x-\mu)/\beta)}\right]_{b}^{\infty} \\
&= 1-\frac{1}{1+e^{-(a-\mu)/\beta}}+\lim_{x\to-\infty}\left(\frac{1}{1+e^{-(x-\mu)/\beta}}\right)-\lim_{x\to\infty}\left(\frac{1}{1+e^{-(x-\mu)/\beta}}\right)+\frac{1}{1+e^{-(b-\mu)/\beta}} \\
&= 1-\frac{1}{1+e^{-(a-\mu)/\beta}}+0-1+\frac{1}{1+e^{-(b-\mu)/\beta}}.
\end{aligned}
$$

Therefore,

$$
\eta=\frac{1}{1+e^{-(b-\mu)/\beta}}-\frac{1}{1+e^{-(a-\mu)/\beta}}\ . \tag{D.12}
$$

Thus, (D.11) is equivalent to

$$
f_t(x)=\frac{\dfrac{e^{-(x-\mu)/\beta}}{\beta\cdot(1+e^{-(x-\mu)/\beta})^2}}{\dfrac{1}{1+e^{-(b-\mu)/\beta}}-\dfrac{1}{1+e^{-(a-\mu)/\beta}}}=\frac{e^{-(x-\mu)/\beta}}{\eta\beta\cdot(1+e^{-(x-\mu)/\beta})^2}. \tag{D.13}
$$

Next, $F_t(x)$ is calculated by

$$
\begin{aligned}
F_t(x) &= \frac{1}{\eta}\cdot\int_{a}^{x}\frac{e^{-(z-\mu)/\beta}}{\beta\cdot(1+e^{-(z-\mu)/\beta})^2}dz \\
&= \frac{1}{\eta}\cdot\left[\frac{1}{1+e^{-(z-\mu)/\beta}}\right]_{a}^{x} \\
&= \frac{1}{\eta}\cdot\left[\frac{1}{1+e^{-(x-\mu)/\beta}}-\frac{1}{1+e^{-(a-\mu)/\beta}}\right],
\end{aligned} \tag{D.14}
$$

where η is defined in (D.12). Then after some rearrangement,

$$
F_t^{-1}(x)=\mu-\beta\ln\left(\left(\eta R+\frac{1}{1+e^{-(a-\mu)/\beta}}\right)^{-1}-1\right). \tag{D.15}
$$

The VBA code for equations (D.13), (D.14), and (D.15) is provided below.

```
Function TLogPDF(Min, Mean, Max, BetaOverMean, Value)
If Value >= Min And Value <= Max Then
    Beta = Mean * BetaOverMean
    Eta = 1 / (1 / (1 + Exp((Mean - Max) / Beta)) - 1 / (1 + Exp((Mean - Min) /
Beta)))
    TLogPDF = Eta * Exp((Mean - Value) / Beta) / (Beta * (1 + Exp((Mean - Value) /
Beta)) ^ 2)
End If
End Function

Function TLogCDF(Min, Mean, Max, BetaOverMean, Value)
If Value >= Min And Value <= Max Then
    Beta = Mean * BetaOverMean
    Eta = 1 / (1 / (1 + Exp((Mean - Max) / Beta)) - 1 / (1 + Exp((Mean - Min) /
Beta)))
    TLogCDF = Eta * (1 / (1 + Exp((Mean - Value) / Beta)) - 1 / (1 + Exp((Mean - Min)
/ Beta)))
End If
End Function

Function TLogDist(Min, Mean, Max, BetaOverMean, Value)
If Value >= 0 And Value <= 1 Then
    Beta = Mean * BetaOverMean
    Eta = 1 / (1 / (1 + Exp((Mean - Max) / Beta)) - 1 / (1 + Exp((Mean - Min) /
Beta)))
    TLogDist = Mean - Beta * Log(1 / (Value / Eta + 1 / (1 + Exp((Mean - Min) /
Beta))) - 1)
End If
End Function
```

Appendix E

Visual Basic for Applications Source Code

E.1 High Level Model

Simulation Module

```
Dim Conf As Double  'Confidence interval (e.g. 0.9 for 90%)
Dim n As Integer    'Number of simulations
Dim R As Integer    'Row number
Dim C As Integer    'Column number
Dim C_cost() As Double  'Construction cost array
Dim O_cost() As Double  'Operating cost array

Sub Simulate()

'Get the specifications from the Options sheet
    'Display status
        Application.StatusBar = "Getting user options..."
    'Move to Options sheet
        Sheets("Options").Select
    'Find the cell that says "Simulation Options"
        Call FindCell("Simulation Options", 100, 2, R, C)
    'Store the number of simulations to perform
        n = Cells(R + 1, C + 1)
    'Store the confidence interval to report
        Conf = Cells(R + 2, C + 1)

'Find the simulation results on the HiLevModel sheet
    'Display status
        Application.StatusBar = "Finding simulation result cells..."
    'Move to HiLevModel sheet
        Sheets("HiLevModel").Select
    'Store the row and column number of the cell
    'that says "Current simulation results" as R, C
        Call FindCell("Current simulation result", 200, 2, R, C)

'Initialize arrays
    ReDim C_cost(n) As Double
    ReDim O_cost(n) As Double

'Turn calculation to manual
    Application.Calculation = xlManual

'Do simulations
    For i = 1 To n
        'Show progress
            Application.StatusBar = "Simulating (" & Int(i / n * 100) & "%)"
        'Store values
            C_cost(i) = Cells(R + 1, C + 1)
            O_cost(i) = Cells(R + 2, C + 1)
        'Recalculate sheet
            Calculate
    Next i

'Turn calculation to automatic
    Application.Calculation = xlCalculationAutomatic

'Set up a new worksheet for the results
    'Create and name new sheet
        Worksheets.Add
        ActiveSheet.Name = "SimData"
    'Create headings
        'Simulation Results
            Cells(1, 1) = "SIMULATION RESULTS"
```

```
            With Range("A1")
                .Font.Bold = True
                .Font.Size = 14
            End With
        'Sim Number, Construction Cost and Operating Cost
            Cells(2, 1) = "Data"
            Cells(2, 2) = "Construction Cost"
            Cells(2, 3) = "Operating Cost"
            With Range("A2:C2")
                .Font.Bold = True
                .HorizontalAlignment = xlRight
            End With
        'Construction Cost and Operating Cost
            Cells(2, 5) = "Construction Cost"
            Cells(2, 6) = "Operating Cost"
            With Range("A2:C2")
                .Font.Bold = True
                .HorizontalAlignment = xlRight
            End With
        'Chart Data
            Cells(2, 8) = "Construction Cost Distribution"
            Cells(2, 11) = "Operating Cost Distribution"
            Cells(3, 8) = "Cost ($ million)"
            Cells(3, 9) = "Frequency"
            Cells(3, 11) = "Cost ($ million)"
            Cells(3, 12) = "Frequency"
            With Range("H2", "K2")
                .Font.Bold = True
            End With
            With Range("H3:I3")
                .Font.Bold = True
                .HorizontalAlignment = xlRight
            End With
            With Range("K3:L3")
                .Font.Bold = True
                .HorizontalAlignment = xlRight
            End With
    'Size columns
        Columns("B:C").ColumnWidth = 20
        Columns("E:F").ColumnWidth = 20
        Columns("H:I").ColumnWidth = 15
        Columns("K:L").ColumnWidth = 15

'Print the results to the new worksheet (SimData)
    For i = 1 To n
        'Display status
            Application.StatusBar = "Generating results (" & Int(i / n * 100) & "%)"
        'Output results to sheet
            Cells(2 + i, 1) = i
            Cells(2 + i, 2) = C_cost(i)
            Cells(2 + i, 3) = O_cost(i)
    Next i

'Format data
    Range("B3:C3").Select
    Range(Selection, Selection.End(xlDown)).NumberFormat = "#,##0"

'Order the simulation results from least to greatest
    Range("B2").Select
    Range(Selection, Selection.End(xlDown)).Select
    Range(Selection, Selection.End(xlToRight)).Select
    Selection.Copy
    Range("E2").Select
```

```
    ActiveSheet.Paste
    Range("E3").Select
    Range(Selection, Selection.End(xlDown)).Select
    Application.CutCopyMode = False
    Selection.Sort Key1:=Range("E3"), Order1:=xlAscending, Header:=xlGuess, _
        OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom, _
        DataOption1:=xlSortNormal
    Range("F3").Select
    Range(Selection, Selection.End(xlDown)).Select
    Selection.Sort Key1:=Range("F3"), Order1:=xlAscending, Header:=xlGuess, _
        OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom, _
        DataOption1:=xlSortNormal

'Display status
    Application.StatusBar = "Creating frequency tables..."

'Create construction cost frequency table
    For i = 1 To 37
        Cells(3 + i, 8) = 50 + 25 * (i - 1)
        Cells(3 + i, 9).FormulaR1C1 = "=COUNTIF(R3C5:R" & n + 2 & "C5,""<=""&RC[-
1]*1000000)-COUNTIF(R3C5:R" & n + 2 & "C5,""<=""&R[-1]C[-1]*1000000)"
    Next i

'Create operating cost frequency table
    For i = 1 To 28
        Cells(3 + i, 11) = 40 + 5 * (i - 1)
        Cells(3 + i, 12).FormulaR1C1 = "=COUNTIF(R3C6:R" & n + 2 & "C6,""<=""&RC[-
1]*1000000)-COUNTIF(R3C6:R" & n + 2 & "C6,""<=""&R[-1]C[-1]*1000000)"
    Next i

'Display status
    Application.StatusBar = "Creating graphs..."

'Turn off screen updating
    Application.ScreenUpdating = False

'Generate report
    'Create Report worksheet
        Sheets.Add
        ActiveSheet.Name = "Report"
    'Format row height, column width, and vertical alignment
        With Cells
            .RowHeight = 15
            .VerticalAlignment = xlCenter
        End With
        Columns("A:A").ColumnWidth = 35
        Columns("B:D").ColumnWidth = 11
    'Create headings
        Range("A1").FormulaR1C1 = "SIMULATION REPORT"
        With Range("A1")
            .Font.Bold = True
            .Font.Name = "Arial"
            .Font.Size = 14
        End With
        Range("A3").FormulaR1C1 = "Construction Cost Results"
        Range("A3").Font.Bold = True
        Range("A4").FormulaR1C1 = _
            "=""Minimum cost ("" & INT(Options!R13C3*100) & ""% confidence)"""
        Range("A5").FormulaR1C1 = "Expected construction cost"
        Range("A6").FormulaR1C1 = _
            "=""Maximum cost ("" & INT(Options!R13C3*100) & ""% confidence)"""
        Range("C4").FormulaR1C1 = "million"
        Range("C5").FormulaR1C1 = "million"
```

```
        Range("C6").FormulaR1C1 = "million"
        Range("A8").FormulaR1C1 = "Operating Cost Results"
        Range("A8").Font.Bold = True
        Range("A9").FormulaR1C1 = _
            "=""Minimum cost ("" & INT(Options!R[4]C[2]*100) & ""% confidence)"""
        Range("A10").FormulaR1C1 = "Expected operating cost"
        Range("A11").FormulaR1C1 = _
            "=""Maximum cost ("" & INT(Options!R[2]C[2]*100) & ""% confidence)"""
        Range("C9").FormulaR1C1 = "million"
        Range("C10").FormulaR1C1 = "million"
        Range("C11").FormulaR1C1 = "million"
        Range("A13").FormulaR1C1 = "Discounted Construction Costs ($ millions)"
        Range("A13").Font.Bold = True
        Range("A14").FormulaR1C1 = "Discount period (years)"
        Range("B14").FormulaR1C1 = "=Options!R7C3"
        Range("C14").FormulaR1C1 = "=Options!R8C3"
        Range("D14").FormulaR1C1 = "=Options!R9C3"
        Range("A14:D14").Font.Bold = True
        Range("A15").FormulaR1C1 = "Minimum discounted cost per year"
        Range("A16").FormulaR1C1 = "Expected discounted cost per year"
        Range("A17").FormulaR1C1 = "Maximum discounted cost per year"
        Range("A4:A6").InsertIndent 1
        Range("A9:A11").InsertIndent 1
        Range("A15:A17").InsertIndent 1
    'Calculate outputs
        Range("B4").FormulaR1C1 = _
            "=mround(INDEX(SimData!R[-1]C[3]:R[" & n - 2 & "]C[3],(1-
Options!R[9]C[1])/2*Options!R[8]C[1]),1E6)/1E6"
        Range("B5").FormulaR1C1 = _
            "=SUMPRODUCT(SimData!R4C8:R40C8,SimData!R4C9:R40C9)/Options!R12C3"
        Range("B6").FormulaR1C1 = _
            "=mround(INDEX(SimData!R[-3]C[3]:R[" & n - 4 &
"]C[3],(1+Options!R[7]C[1])/2*Options!R[6]C[1]),1E6)/1E6"
        Range("B9").FormulaR1C1 = _
            "=MROUND(INDEX(SimData!R[-6]C[4]:R[" & n - 7 & "]C[4],(1-
Options!R13C3)/2*Options!R12C3),1000000)/1000000"
        Range("B10").FormulaR1C1 = _
            "=SUMPRODUCT(SimData!R4C11:R31C11,SimData!R4C12:R31C12)/Options!R12C3"
        Range("B11").FormulaR1C1 = _
            "=MROUND(INDEX(SimData!R[-8]C[4]:R[" & n - 9 &
"]C[4],(1+Options!R13C3)/2*Options!R12C3),1000000)/1000000"
        Range("B15").FormulaR1C1 = _
            "=R[-11]C2/R14C*(1-EXP(-inf*R14C))/(1-EXP(-inf))"
        Range("B15").Select
        Selection.AutoFill Destination:=Range("B15:D15"), Type:=xlFillDefault
        Range("B15:D15").Select
        Selection.AutoFill Destination:=Range("B15:D17"), Type:=xlFillDefault
    'Format outputs
        Range("B4:B6").NumberFormat = "$#,##0"
        Range("B9:B11").NumberFormat = "$#,##0"
        Range("B15:D17").NumberFormat = "$#,##0"

'Create construction cost chart
    Range("I4").Select
    Range(Selection, Selection.End(xlDown)).Select
    Charts.Add

'Format construction cost chart
    With ActiveChart
        'Type
            .ChartType = xlColumnClustered
        'Source data
```

```
            .SetSourceData Source:=Sheets("SimData").Range("I4:I40"),
PlotBy:=xlColumns
            .SeriesCollection(1).XValues = "=SimData!R4C8:R40C8"
            .SeriesCollection(1).Values = "=SimData!R4C9:R40C9"
            .SeriesCollection(1).Name = "Construction Cost"
        'Create new sheet for chart and name ConstCostDist
            .Location Where:=xlLocationAsNewSheet, Name:="ConstCostDist"
        'Use white background and no legend
            .PlotArea.Interior.ColorIndex = 2
            .HasLegend = False
        'Chart title
            .HasTitle = True
            .ChartTitle.Characters.Text = "Construction Cost Distribution"
            .ChartTitle.Font.Bold = True
            .ChartTitle.Font.Size = 14
        'Axes
            .Axes(xlCategory, xlPrimary).HasTitle = True
            .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "Construction
Cost ($ millions)"
            .Axes(xlCategory).HasMajorGridlines = False
            .Axes(xlCategory).HasMinorGridlines = False
            .Axes(xlCategory).CrossesAt = 1
            .Axes(xlCategory).TickLabelSpacing = 2
            .Axes(xlCategory).TickMarkSpacing = 1
            .Axes(xlCategory).AxisBetweenCategories = True
            .Axes(xlCategory).ReversePlotOrder = False
            .Axes(xlValue, xlPrimary).HasTitle = True
            .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Frequency"
            .Axes(xlValue).HasMajorGridlines = False
            .Axes(xlValue).HasMinorGridlines = False
        'Plot area border style
            .PlotArea.Border.ColorIndex = 16
            .PlotArea.Border.Weight = xlThin
            .PlotArea.Border.LineStyle = xlContinuous
        'Histogram style
            .ChartGroups(1).Overlap = 0
            .ChartGroups(1).GapWidth = 0
            .ChartGroups(1).HasSeriesLines = False
            .ChartGroups(1).VaryByCategories = False
    End With

'Create operating cost chart
    Sheets("SimData").Select
    Range("L4").Select
    Range(Selection, Selection.End(xlDown)).Select
    Charts.Add

'Format construction cost chart
    With ActiveChart
        'Type
            .ChartType = xlColumnClustered
        'Source data
            .SetSourceData Source:=Sheets("SimData").Range("L4:L31"),
PlotBy:=xlColumns
            .SeriesCollection(1).XValues = "=SimData!R4C11:R31C11"
            .SeriesCollection(1).Values = "=SimData!R4C12:R31C12"
            .SeriesCollection(1).Name = "Operating Cost"
        'Create new sheet for chart and name ConstCostDist
            .Location Where:=xlLocationAsNewSheet, Name:="OperCostDist"
        'Use white background and no legend
            .PlotArea.Interior.ColorIndex = 2
            .HasLegend = False
        'Chart title
```

```
            .HasTitle = True
            .ChartTitle.Characters.Text = "Operating Cost Distribution"
            .ChartTitle.Font.Bold = True
            .ChartTitle.Font.Size = 14
        'Axes
            .Axes(xlCategory, xlPrimary).HasTitle = True
            .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "Operating Cost
($ millions)"
            .Axes(xlCategory).HasMajorGridlines = False
            .Axes(xlCategory).HasMinorGridlines = False
            .Axes(xlCategory).CrossesAt = 1
            .Axes(xlCategory).TickLabelSpacing = 2
            .Axes(xlCategory).TickMarkSpacing = 1
            .Axes(xlCategory).AxisBetweenCategories = True
            .Axes(xlCategory).ReversePlotOrder = False
            .Axes(xlValue, xlPrimary).HasTitle = True
            .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Frequency"
            .Axes(xlValue).HasMajorGridlines = False
            .Axes(xlValue).HasMinorGridlines = False
        'Plot area border style
            .PlotArea.Border.ColorIndex = 16
            .PlotArea.Border.Weight = xlThin
            .PlotArea.Border.LineStyle = xlContinuous
        'Histogram style
            .ChartGroups(1).Overlap = 0
            .ChartGroups(1).GapWidth = 0
            .ChartGroups(1).HasSeriesLines = False
            .ChartGroups(1).VaryByCategories = False
    End With

'Turn off status bar flags and turn on screen updating
    With Application
        .StatusBar = False
        .ScreenUpdating = True
    End With

'Move to report sheet
    Sheets("Report").Select
    Range("A1").Select

End Sub
```

UtilitySubs Module

```
Sub FindCell(search_label As String, row_bound As Integer, col_bound As Integer, ByRef
R As Integer, ByRef C As Integer)

'This is a simple linear search subroutine which is
'called from within other macros to find a certain
'cell on the spreadsheet.
'Inputs:
'search_label = the string (text) appearing in the cell to be found
'row_bound = the last row to search in (look in rows 1 through row_bound)
'col_bound = the last column to search in (look in columns 2 through col_bound)
'Outputs:
'R = the row of the cell containing search_label
'C = the column of the cell containing search_label

Dim Found As Boolean
Dim Loops As Integer
Loops = 0
```

```
Do While Not Found
    'Count loops
    Loops = Loops + 1
    'Check each column two and onward to col_bound
    For col = 2 To col_bound
        'Check each row
        For Row = 1 To row_bound
            If Cells(Row, col) = search_label Then
                'Mark as found
                    Found = True
                'Store the R and C
                    R = Row
                    C = col
            End If
            If Found = True Then
                Exit For
            End If
        Next Row
        If Found = True Then
            Exit For
        End If
    Next col
    'Get out if too many loops
    If Loops > row_bound * col_bound Then
        Exit Do
    End If
Loop

End Sub



Sub Checkbox()

'INPUTS
'Column with previous state of checkboxes
    col_pre_state = 1
'Column with current state of checkboxes
    col_cur_state = 11
'Column with headings of each section
    heading_column = 2
'END INPUTS

'Turn off screen updating and turn on automatic sheet calculation
    Application.ScreenUpdating = False

'Initialize Found
    Dim Found As Boolean
    Found = False

'Initialize row counter
    row_index = 1

'Find the checkbox that changed
Do While Found = False
    'Check the current row
    'See if both checkbox state indicating cells are nonzero and not the same
    If Cells(row_index, col_pre_state) <> Cells(row_index, col_cur_state) And
Cells(row_index, col_pre_state) <> 0 And Cells(row_index, col_cur_state) <> 0 Then
        'Mark as found
        Found = True
        'The first row that needs to be hidden or unhidden is
```

```
        row_start = row_index + 1
    Else
        'Check the next row on the next loop
        row_index = row_index + 1
    End If
    'Get out if too many loops
    If row_index > 10000 Then
        Exit Do
    End If
Loop

'Find the last row that needs to be hidden or unhidden (look for blanks)
Found = False
row_index = row_start + 1

Do While Found = False
    'Check the current row
    'See if the heading column is blank
    If Cells(row_index, heading_column) = 0 Then
        'Mark as found
        Found = True
        'The last row that needs to be hidden or unhidden is
        row_stop = row_index
    Else
        'Check the next row on the next loop
        row_index = row_index + 1
    End If
    'Get out if too many loops
    If row_index > 10000 Then
        Exit Do
    End If
Loop

'Stop if no rows to hide or show
    If row_start = Empty Then
        Exit Sub
    End If

'Decide whether the cells need to be hidden or unhidden
Dim Hidden As Boolean
If Cells(row_start - 1, col_cur_state) = 2 Then
    'Show = false so Hidden = True
    Hidden = True
ElseIf Cells(row_start - 1, col_cur_state) = 1 Then
    'Show = true so Hidden = False
    Hidden = False
Else
    'Something went wrong
    Exit Sub
End If

'Either hide or unhide the rows
Rows(row_start & ":" & row_stop).EntireRow.Hidden = Hidden

'Copy the current state of the checkbox to the previous state
Cells(row_start - 1, col_pre_state) = Cells(row_start - 1, col_cur_state)

'Turn on screen updating and turn off automatic sheet calculation
    With Application
        .ScreenUpdating = True
    End With

End Sub
```

```
Sub GroupCheckbox()

Dim Found As Boolean
Dim Hidden As Boolean
Dim row_start() As Integer
Dim row_end() As Integer
Found = False

'Column with previous state of checkboxes
col_pre_state = 1
'Column with current state of checkboxes
col_cur_state = 11
'Column with TRUE/FALSE checkbox linked cells
col_TF = 12
'Column with number of subboxes
col_subboxes = 13
'Column with headings of each section
heading_column = 2
'Initialize row counter
row_index = 1

'Turn off screen updating and turn on automatic sheet calculation
    With Application
        .ScreenUpdating = False
    End With

'Find the checkbox that changed
Do While Found = False
    'Check the current row
    'See if both checkbox state indicating cells are nonzero and not the same
    If Cells(row_index, col_pre_state) <> Cells(row_index, col_cur_state) And
Cells(row_index, col_pre_state) <> 0 And Cells(row_index, col_cur_state) <> 0 Then
        'Mark as found
        Found = True
        'Get the number of sub checkboxes
        sub_boxes = Cells(row_index, col_subboxes)
    Else
        'Check the next row on the next loop
        row_index = row_index + 1
    End If
    'Get out if too many loops
    If row_index > 10000 Then
        Exit Do
    End If
Loop

'Toggle pre_state value
    If Cells(row_index, col_pre_state) = 2 Then
        Cells(row_index, col_pre_state) = 1
        Hidden = False
    ElseIf Cells(row_index, col_pre_state) = 1 Then
        Cells(row_index, col_pre_state) = 2
        Hidden = True
    Else
        Exit Sub
    End If

'Process all subsections
    'Initialize box counter
        changed_boxes = 0
    'Find each and store cell locations
        Do While changed_boxes < sub_boxes
```

```
            'Check the row after the label
                row_index = row_index + 1
            If Cells(row_index, col_pre_state) <> 0 Then
                'CONCLUSION: row with a subbox found
                'Increase number of boxes changed
                    changed_boxes = changed_boxes + 1
                'Change T/F value
                    Cells(row_index, col_TF) = Not Hidden
                'Change prestate value
                    If Hidden = True Then
                        Cells(row_index, col_pre_state) = 2
                    ElseIf Hidden = False Then
                        Cells(row_index, col_pre_state) = 1
                    End If
                'Store cell block start
                    ReDim Preserve row_start(changed_boxes)
                    row_start(changed_boxes) = row_index + 1
                'Store previous cell block end
                    If changed_boxes > 1 And changed_boxes <= sub_boxes Then
                        ReDim Preserve row_end(changed_boxes - 1)
                        row_end(changed_boxes - 1) = row_index - 1
                    End If
            End If
        Loop
    'Find the end of the last section
        Found = False
        Do While Found = False
            row_index = row_index + 1
            If Cells(row_index, heading_column) = 0 Then
                Found = True
                ReDim Preserve row_end(changed_boxes)
                row_end(changed_boxes) = row_index
            End If
        Loop

'Hide or unhide all the cells
For i = 1 To sub_boxes
    Rows(row_start(i) & ":" & row_end(i)).EntireRow.Hidden = Hidden
Next i

'Turn on screen updating and (turn off automatic sheet calculation)
    Application.ScreenUpdating = True

End Sub
```

E.2 Detailed Model

PublicVariables Module

```
'Row and column variables
Public R As Integer
Public C As Integer

'Main form options
Public FrmMain_optUseAssistance As Boolean
Public FrmMain_optByMyself As Boolean

'Create New Model form options
Public FrmCreateNew_optCreateNew As Boolean
Public FrmCreateNew_optStartOver As Boolean
```

```
'Forms
Public Max_item_count As Integer    'Largest number of items allowed in any category
Public Max_cat_count As Integer     'Largest number of categories allowed
Public Cat() As String          'Names of each category
Public Item() As String         'Names of each type of facility
Public Item_old() As String     'Names of each type of facility (placeholder)
Public Qty() As String         'Quantity of each item
Public Size() As String     'Size of each item
Public Price() As String        'Cost per square foot of each item
Public Cat_count As Integer     'Number of categories
Public Cat_num As Integer       'Category that user is currently modifying
Public Item_count() As Integer    'Number of line items
Public Form_num As Integer      'Form that user is currently modifying

'Flags
Public Cancelled As Boolean     'True when user confirms cancellation of model
creation
Public Found As Boolean         'Search result indicator
```

UtilitySubs Module

```
Sub Init_frmFacility(Param_cat_num As Integer, Param_form_num As Integer)

'Transfer parameter values
    Cat_num = Param_cat_num
    Form_num = Param_form_num

'Category n of N heading
    frmFacility.frmFacility_lbl_CategoryN.Caption = "Category " & Cat_num & " of " &
Cat_count
'Category name
    frmFacility.frmFacility_txt_CategoryName.Value = Cat(Cat_num)
'Form n of N heading
    msg = "This category currently has " & Item_count(Cat_num) &
IIf(Item_count(Cat_num) <> 1, " Facilities", " Facility") & ", displayed on "
    msg = msg & Int(Item_count(Cat_num) / 10) + 1 & IIf(Item_count(Cat_num) > 10, "
Forms.  ", " Form.  ")
    msg = msg & "This is Form " & Form_num & " of " & Int(Item_count(Cat_num) / 10) +
1
    msg = msg & ", displaying Facilities " & 10 * Form_num - 9 & " - "
    msg = msg & IIf(Item_count(Cat_num) < 10 * Form_num, Item_count(Cat_num), 10 *
Form_num)
    msg = msg & " of " & Item_count(Cat_num) & "."
    frmFacility.frmFacility_lbl_FormN.Caption = msg
'Field numbers
    frmFacility.frmFacility_lbl_Number01.Caption = 10 * Form_num - 9
    frmFacility.frmFacility_lbl_Number02.Caption = 10 * Form_num - 8
    frmFacility.frmFacility_lbl_Number03.Caption = 10 * Form_num - 7
    frmFacility.frmFacility_lbl_Number04.Caption = 10 * Form_num - 6
    frmFacility.frmFacility_lbl_Number05.Caption = 10 * Form_num - 5
    frmFacility.frmFacility_lbl_Number06.Caption = 10 * Form_num - 4
    frmFacility.frmFacility_lbl_Number07.Caption = 10 * Form_num - 3
    frmFacility.frmFacility_lbl_Number08.Caption = 10 * Form_num - 2
    frmFacility.frmFacility_lbl_Number09.Caption = 10 * Form_num - 1
    frmFacility.frmFacility_lbl_Number10.Caption = 10 * Form_num
'Facility names
    frmFacility.frmFacility_txt_Name01.Value = Item(Cat_num,
frmFacility.frmFacility_lbl_Number01.Caption)
    frmFacility.frmFacility_txt_Name02.Value = Item(Cat_num,
frmFacility.frmFacility_lbl_Number02.Caption)
```

```
    frmFacility.frmFacility_txt_Name03.Value = Item(Cat_num, 
frmFacility.frmFacility_lbl_Number03.Caption)
    frmFacility.frmFacility_txt_Name04.Value = Item(Cat_num, 
frmFacility.frmFacility_lbl_Number04.Caption)
    frmFacility.frmFacility_txt_Name05.Value = Item(Cat_num, 
frmFacility.frmFacility_lbl_Number05.Caption)
    frmFacility.frmFacility_txt_Name06.Value = Item(Cat_num, 
frmFacility.frmFacility_lbl_Number06.Caption)
    frmFacility.frmFacility_txt_Name07.Value = Item(Cat_num, 
frmFacility.frmFacility_lbl_Number07.Caption)
    frmFacility.frmFacility_txt_Name08.Value = Item(Cat_num, 
frmFacility.frmFacility_lbl_Number08.Caption)
    frmFacility.frmFacility_txt_Name09.Value = Item(Cat_num, 
frmFacility.frmFacility_lbl_Number09.Caption)
    frmFacility.frmFacility_txt_Name10.Value = Item(Cat_num, 
frmFacility.frmFacility_lbl_Number10.Caption)
'Quantities
    frmFacility.frmFacility_txt_Qty01.Value = IIf(Qty(Cat_num, 
frmFacility.frmFacility_lbl_Number01.Caption) <> "", Qty(Cat_num, 
frmFacility.frmFacility_lbl_Number01.Caption), "")
    frmFacility.frmFacility_txt_Qty02.Value = IIf(Qty(Cat_num, 
frmFacility.frmFacility_lbl_Number02.Caption) <> "", Qty(Cat_num, 
frmFacility.frmFacility_lbl_Number02.Caption), "")
    frmFacility.frmFacility_txt_Qty03.Value = IIf(Qty(Cat_num, 
frmFacility.frmFacility_lbl_Number03.Caption) <> "", Qty(Cat_num, 
frmFacility.frmFacility_lbl_Number03.Caption), "")
    frmFacility.frmFacility_txt_Qty04.Value = IIf(Qty(Cat_num, 
frmFacility.frmFacility_lbl_Number04.Caption) <> "", Qty(Cat_num, 
frmFacility.frmFacility_lbl_Number04.Caption), "")
    frmFacility.frmFacility_txt_Qty05.Value = IIf(Qty(Cat_num, 
frmFacility.frmFacility_lbl_Number05.Caption) <> "", Qty(Cat_num, 
frmFacility.frmFacility_lbl_Number05.Caption), "")
    frmFacility.frmFacility_txt_Qty06.Value = IIf(Qty(Cat_num, 
frmFacility.frmFacility_lbl_Number06.Caption) <> "", Qty(Cat_num, 
frmFacility.frmFacility_lbl_Number06.Caption), "")
    frmFacility.frmFacility_txt_Qty07.Value = IIf(Qty(Cat_num, 
frmFacility.frmFacility_lbl_Number07.Caption) <> "", Qty(Cat_num, 
frmFacility.frmFacility_lbl_Number07.Caption), "")
    frmFacility.frmFacility_txt_Qty08.Value = IIf(Qty(Cat_num, 
frmFacility.frmFacility_lbl_Number08.Caption) <> "", Qty(Cat_num, 
frmFacility.frmFacility_lbl_Number08.Caption), "")
    frmFacility.frmFacility_txt_Qty09.Value = IIf(Qty(Cat_num, 
frmFacility.frmFacility_lbl_Number09.Caption) <> "", Qty(Cat_num, 
frmFacility.frmFacility_lbl_Number09.Caption), "")
    frmFacility.frmFacility_txt_Qty10.Value = IIf(Qty(Cat_num, 
frmFacility.frmFacility_lbl_Number10.Caption) <> "", Qty(Cat_num, 
frmFacility.frmFacility_lbl_Number10.Caption), "")
'Sizes
    frmFacility.frmFacility_txt_Size01.Value = IIf(Size(Cat_num, 
frmFacility.frmFacility_lbl_Number01.Caption) <> "", Size(Cat_num, 
frmFacility.frmFacility_lbl_Number01.Caption), "")
    frmFacility.frmFacility_txt_Size02.Value = IIf(Size(Cat_num, 
frmFacility.frmFacility_lbl_Number02.Caption) <> "", Size(Cat_num, 
frmFacility.frmFacility_lbl_Number02.Caption), "")
    frmFacility.frmFacility_txt_Size03.Value = IIf(Size(Cat_num, 
frmFacility.frmFacility_lbl_Number03.Caption) <> "", Size(Cat_num, 
frmFacility.frmFacility_lbl_Number03.Caption), "")
    frmFacility.frmFacility_txt_Size04.Value = IIf(Size(Cat_num, 
frmFacility.frmFacility_lbl_Number04.Caption) <> "", Size(Cat_num, 
frmFacility.frmFacility_lbl_Number04.Caption), "")
    frmFacility.frmFacility_txt_Size05.Value = IIf(Size(Cat_num, 
frmFacility.frmFacility_lbl_Number05.Caption) <> "", Size(Cat_num, 
frmFacility.frmFacility_lbl_Number05.Caption), "")
```

```
    frmFacility.frmFacility_txt_Size06.Value = IIf(Size(Cat_num,
frmFacility.frmFacility_lbl_Number06.Caption) <> "", Size(Cat_num,
frmFacility.frmFacility_lbl_Number06.Caption), "")
    frmFacility.frmFacility_txt_Size07.Value = IIf(Size(Cat_num,
frmFacility.frmFacility_lbl_Number07.Caption) <> "", Size(Cat_num,
frmFacility.frmFacility_lbl_Number07.Caption), "")
    frmFacility.frmFacility_txt_Size08.Value = IIf(Size(Cat_num,
frmFacility.frmFacility_lbl_Number08.Caption) <> "", Size(Cat_num,
frmFacility.frmFacility_lbl_Number08.Caption), "")
    frmFacility.frmFacility_txt_Size09.Value = IIf(Size(Cat_num,
frmFacility.frmFacility_lbl_Number09.Caption) <> "", Size(Cat_num,
frmFacility.frmFacility_lbl_Number09.Caption), "")
    frmFacility.frmFacility_txt_Size10.Value = IIf(Size(Cat_num,
frmFacility.frmFacility_lbl_Number10.Caption) <> "", Size(Cat_num,
frmFacility.frmFacility_lbl_Number10.Caption), "")
'Prices
    frmFacility.frmFacility_txt_Price01.Value = IIf(Price(Cat_num,
frmFacility.frmFacility_lbl_Number01.Caption) <> "", Price(Cat_num,
frmFacility.frmFacility_lbl_Number01.Caption), "")
    frmFacility.frmFacility_txt_Price02.Value = IIf(Price(Cat_num,
frmFacility.frmFacility_lbl_Number02.Caption) <> "", Price(Cat_num,
frmFacility.frmFacility_lbl_Number02.Caption), "")
    frmFacility.frmFacility_txt_Price03.Value = IIf(Price(Cat_num,
frmFacility.frmFacility_lbl_Number03.Caption) <> "", Price(Cat_num,
frmFacility.frmFacility_lbl_Number03.Caption), "")
    frmFacility.frmFacility_txt_Price04.Value = IIf(Price(Cat_num,
frmFacility.frmFacility_lbl_Number04.Caption) <> "", Price(Cat_num,
frmFacility.frmFacility_lbl_Number04.Caption), "")
    frmFacility.frmFacility_txt_Price05.Value = IIf(Price(Cat_num,
frmFacility.frmFacility_lbl_Number05.Caption) <> "", Price(Cat_num,
frmFacility.frmFacility_lbl_Number05.Caption), "")
    frmFacility.frmFacility_txt_Price06.Value = IIf(Price(Cat_num,
frmFacility.frmFacility_lbl_Number06.Caption) <> "", Price(Cat_num,
frmFacility.frmFacility_lbl_Number06.Caption), "")
    frmFacility.frmFacility_txt_Price07.Value = IIf(Price(Cat_num,
frmFacility.frmFacility_lbl_Number07.Caption) <> "", Price(Cat_num,
frmFacility.frmFacility_lbl_Number07.Caption), "")
    frmFacility.frmFacility_txt_Price08.Value = IIf(Price(Cat_num,
frmFacility.frmFacility_lbl_Number08.Caption) <> "", Price(Cat_num,
frmFacility.frmFacility_lbl_Number08.Caption), "")
    frmFacility.frmFacility_txt_Price09.Value = IIf(Price(Cat_num,
frmFacility.frmFacility_lbl_Number09.Caption) <> "", Price(Cat_num,
frmFacility.frmFacility_lbl_Number09.Caption), "")
    frmFacility.frmFacility_txt_Price10.Value = IIf(Price(Cat_num,
frmFacility.frmFacility_lbl_Number10.Caption) <> "", Price(Cat_num,
frmFacility.frmFacility_lbl_Number10.Caption), "")
'Costs
    'frmFacility.frmFacility_lbl_Cost01.Caption = IIf(Qty(Cat_num,
frmFacility.frmFacility_lbl_Number01.Caption) * Size(Cat_num,
frmFacility.frmFacility_lbl_Number01.Caption) * Price(Cat_num,
frmFacility.frmFacility_lbl_Number01.Caption) > 0, Qty(Cat_num,
frmFacility.frmFacility_lbl_Number01.Caption) * Size(Cat_num,
frmFacility.frmFacility_lbl_Number01.Caption) * Price(Cat_num,
frmFacility.frmFacility_lbl_Number01.Caption), "")
    'frmFacility.frmFacility_lbl_Cost02.Caption = IIf(Qty(Cat_num,
frmFacility.frmFacility_lbl_Number02.Caption) * Size(Cat_num,
frmFacility.frmFacility_lbl_Number02.Caption) * Price(Cat_num,
frmFacility.frmFacility_lbl_Number02.Caption) > 0, Qty(Cat_num,
frmFacility.frmFacility_lbl_Number02.Caption) * Size(Cat_num,
frmFacility.frmFacility_lbl_Number02.Caption) * Price(Cat_num,
frmFacility.frmFacility_lbl_Number02.Caption), "")
    'frmFacility.frmFacility_lbl_Cost03.Caption = IIf(Qty(Cat_num,
frmFacility.frmFacility_lbl_Number03.Caption) * Size(Cat_num,
```

```
frmFacility.frmFacility_lbl_Number03.Caption) * Price(Cat_num,
frmFacility.frmFacility_lbl_Number03.Caption) > 0, Qty(Cat_num,
frmFacility.frmFacility_lbl_Number03.Caption) * Size(Cat_num,
frmFacility.frmFacility_lbl_Number03.Caption) * Price(Cat_num,
frmFacility.frmFacility_lbl_Number03.Caption), "")
    'frmFacility.frmFacility_lbl_Cost04.Caption = IIf(Qty(Cat_num,
frmFacility.frmFacility_lbl_Number04.Caption) * Size(Cat_num,
frmFacility.frmFacility_lbl_Number04.Caption) * Price(Cat_num,
frmFacility.frmFacility_lbl_Number04.Caption) > 0, Qty(Cat_num,
frmFacility.frmFacility_lbl_Number04.Caption) * Size(Cat_num,
frmFacility.frmFacility_lbl_Number04.Caption) * Price(Cat_num,
frmFacility.frmFacility_lbl_Number04.Caption), "")
    'frmFacility.frmFacility_lbl_Cost05.Caption = IIf(Qty(Cat_num,
frmFacility.frmFacility_lbl_Number05.Caption) * Size(Cat_num,
frmFacility.frmFacility_lbl_Number05.Caption) * Price(Cat_num,
frmFacility.frmFacility_lbl_Number05.Caption) > 0, Qty(Cat_num,
frmFacility.frmFacility_lbl_Number05.Caption) * Size(Cat_num,
frmFacility.frmFacility_lbl_Number05.Caption) * Price(Cat_num,
frmFacility.frmFacility_lbl_Number05.Caption), "")
    'frmFacility.frmFacility_lbl_Cost06.Caption = IIf(Qty(Cat_num,
frmFacility.frmFacility_lbl_Number06.Caption) * Size(Cat_num,
frmFacility.frmFacility_lbl_Number06.Caption) * Price(Cat_num,
frmFacility.frmFacility_lbl_Number06.Caption) > 0, Qty(Cat_num,
frmFacility.frmFacility_lbl_Number06.Caption) * Size(Cat_num,
frmFacility.frmFacility_lbl_Number06.Caption) * Price(Cat_num,
frmFacility.frmFacility_lbl_Number06.Caption), "")
    'frmFacility.frmFacility_lbl_Cost07.Caption = IIf(Qty(Cat_num,
frmFacility.frmFacility_lbl_Number07.Caption) * Size(Cat_num,
frmFacility.frmFacility_lbl_Number07.Caption) * Price(Cat_num,
frmFacility.frmFacility_lbl_Number07.Caption) > 0, Qty(Cat_num,
frmFacility.frmFacility_lbl_Number07.Caption) * Size(Cat_num,
frmFacility.frmFacility_lbl_Number07.Caption) * Price(Cat_num,
frmFacility.frmFacility_lbl_Number07.Caption), "")
    'frmFacility.frmFacility_lbl_Cost08.Caption = IIf(Qty(Cat_num,
frmFacility.frmFacility_lbl_Number08.Caption) * Size(Cat_num,
frmFacility.frmFacility_lbl_Number08.Caption) * Price(Cat_num,
frmFacility.frmFacility_lbl_Number08.Caption) > 0, Qty(Cat_num,
frmFacility.frmFacility_lbl_Number08.Caption) * Size(Cat_num,
frmFacility.frmFacility_lbl_Number08.Caption) * Price(Cat_num,
frmFacility.frmFacility_lbl_Number08.Caption), "")
    'frmFacility.frmFacility_lbl_Cost09.Caption = IIf(Qty(Cat_num,
frmFacility.frmFacility_lbl_Number09.Caption) * Size(Cat_num,
frmFacility.frmFacility_lbl_Number09.Caption) * Price(Cat_num,
frmFacility.frmFacility_lbl_Number09.Caption) > 0, Qty(Cat_num,
frmFacility.frmFacility_lbl_Number09.Caption) * Size(Cat_num,
frmFacility.frmFacility_lbl_Number09.Caption) * Price(Cat_num,
frmFacility.frmFacility_lbl_Number09.Caption), "")
    'frmFacility.frmFacility_lbl_Cost10.Caption = IIf(Qty(Cat_num,
frmFacility.frmFacility_lbl_Number10.Caption) * Size(Cat_num,
frmFacility.frmFacility_lbl_Number10.Caption) * Price(Cat_num,
frmFacility.frmFacility_lbl_Number10.Caption) > 0, Qty(Cat_num,
frmFacility.frmFacility_lbl_Number10.Caption) * Size(Cat_num,
frmFacility.frmFacility_lbl_Number10.Caption) * Price(Cat_num,
frmFacility.frmFacility_lbl_Number10.Caption), "")
'Clear buttons
    frmFacility.frmFacility_btn_Clear01.Enabled =
IIf(frmFacility.frmFacility_txt_Name01.Value <> "", True, False)
    frmFacility.frmFacility_btn_Clear02.Enabled =
IIf(frmFacility.frmFacility_txt_Name02.Value <> "", True, False)
    frmFacility.frmFacility_btn_Clear03.Enabled =
IIf(frmFacility.frmFacility_txt_Name03.Value <> "", True, False)
    frmFacility.frmFacility_btn_Clear04.Enabled =
IIf(frmFacility.frmFacility_txt_Name04.Value <> "", True, False)
```

```
    frmFacility.frmFacility_btn_Clear05.Enabled =
IIf(frmFacility.frmFacility_txt_Name05.Value <> "", True, False)
    frmFacility.frmFacility_btn_Clear06.Enabled =
IIf(frmFacility.frmFacility_txt_Name06.Value <> "", True, False)
    frmFacility.frmFacility_btn_Clear07.Enabled =
IIf(frmFacility.frmFacility_txt_Name07.Value <> "", True, False)
    frmFacility.frmFacility_btn_Clear08.Enabled =
IIf(frmFacility.frmFacility_txt_Name08.Value <> "", True, False)
    frmFacility.frmFacility_btn_Clear09.Enabled =
IIf(frmFacility.frmFacility_txt_Name09.Value <> "", True, False)
    frmFacility.frmFacility_btn_Clear10.Enabled =
IIf(frmFacility.frmFacility_txt_Name10.Value <> "", True, False)
'Category navigation buttons
    frmFacility.frmFacility_btn_FirstCat.Enabled = IIf(Cat_num > 1, True, False)
    frmFacility.frmFacility_btn_PrevCat.Enabled = IIf(Cat_num > 1, True, False)
    frmFacility.frmFacility_btn_NextCat.Enabled = IIf(Cat_num < Cat_count, True,
False)
    frmFacility.frmFacility_btn_LastCat.Enabled = IIf(Cat_num < Cat_count, True,
False)
'Form navigation buttons
    frmFacility.frmFacility_btn_FirstForm.Enabled = IIf(Form_num > 1, True, False)
    frmFacility.frmFacility_btn_PrevForm.Enabled = IIf(Form_num > 1, True, False)
    frmFacility.frmFacility_btn_NextForm.Enabled = IIf(Form_num <
Int(Item_count(Cat_num) / 10) + 1, True, False)
    frmFacility.frmFacility_btn_LastForm.Enabled = IIf(Form_num <
Int(Item_count(Cat_num) / 10) + 1, True, False)
'More Blank Fields, Clear Entire Form, Clear Entire Category buttons

End Sub


Sub FindCell(search_label As String, row_bound As Integer, col_bound As Integer, ByRef
R As Integer, ByRef C As Integer)

Dim Found As Boolean
Dim Loops As Integer
Loops = 0

Do While Not Found
    'Count loops
    Loops = Loops + 1
    'Check each column two and onward to col_bound
    For col = 1 To col_bound
        'Check each row
        For Row = 1 To row_bound
            If Cells(Row, col) = search_label Then
                'Mark as found
                    Found = True
                'Store the R and C
                    R = Row
                    C = col
            End If
            If Found = True Then
                Exit For
            End If
        Next Row
        If Found = True Then
            Exit For
        End If
    Next col
    'Get out if too many loops
    If Loops > row_bound * col_bound Then
        MsgBox "The search text (" & search_label & ") could not be found."
```

```
        Exit Do
    End If
Loop

End Sub

Sub SaveFile()

'Display file creation message
    msg = "First you will create a new Excel file for your new model." & vbCrLf
    msg = msg & vbCrLf & "You must pick a different file name (other than
Detailed_Model)" & vbCrLf
    msg = msg & "so that the original file is preserved." & vbCrLf
    msg = msg & vbCrLf & "For example, you might want to name the new file
Detailed_Model_01 or Bobs_Model." & vbCrLf
    msg = msg & vbCrLf & "Click OK to proceed."
    MsgBox msg, vbOKOnly, "File Creation"
'Prompt for new file name
    fname = Application.GetSaveAsFilename
'Abort if user clicks Cancel
    If fname = False Then
        MsgBox "File creation was cancelled."
        Exit Sub
    End If
'Append the .xls extension if necessary
    If Right(fname, 4) <> ".xls" Then
        fname = fname & "xls"
    End If
'Request user to change the file name if they pick the same name
    Do While fname = ActiveWorkbook.FullName And fname <> False
        MsgBox "You must choose a different file name."
        fname = Application.GetSaveAsFilename
        'Abort if user clicks Cancel
            If fname = False Then
                MsgBox "File creation was cancelled."
                Exit Sub
            End If
        'Append the .xls extension if necessary
            If Right(fname, 4) <> ".xls" Then
                fname = fname & "xls"
            End If
    Loop
'Save file with new name
    ActiveWorkbook.SaveAs fname

End Sub


Sub OpenFacilityForm()

'Find cell with heading "Facility Cost"
    Worksheets("Construction Cost").Select
    Call FindCell("Facility Cost", 1000, 2, R, C)

'Initialization
    Done = False
    R = R + 1
    Cat_count = 0
    Max_cat_count = 100
    Max_item_count = 100

'Assume sufficient sizes for arrays
    ReDim Item(Max_cat_count, Max_item_count)
```

```
    ReDim Qty(Max_cat_count, Max_item_count)
    ReDim Size(Max_cat_count, Max_item_count)
    ReDim Price(Max_cat_count, Max_item_count)
    ReDim Cat(Max_cat_count)
    ReDim Item_count(Max_item_count)

'Store all fields
    Do While Done = False

    'Check two rows down for Category
        If Left(Cells(R + 2, C), 8) = "Category" Then
            'Move to row
                R = R + 2
            'Count categories
                Cat_count = Cat_count + 1
            'Store category name
                Cat(Cat_count) = Right(Cells(R, C), Len(Cells(R, C)) - InStr(Cells(R,
C), "-") - 1)
            'Check next row down for SUBTOTAL
                Do While Left(Cells(R + 1, C), 8) <> "SUBTOTAL"
                    'Move to row
                        R = R + 1
                    'Count items in the category
                        Item_count(Cat_count) = Item_count(Cat_count) + 1
                        Item(Cat_count, Item_count(Cat_count)) = Cells(R,
C).FormulaR1C1
                        Qty(Cat_count, Item_count(Cat_count)) = Cells(R, C +
1).FormulaR1C1
                        Size(Cat_count, Item_count(Cat_count)) = Cells(R, C +
2).FormulaR1C1
                        Price(Cat_count, Item_count(Cat_count)) = Cells(R, C +
3).FormulaR1C1
                Loop
            'Move to row
                R = R + 1
        Else
            'Flag as done
                Done = True
        End If

    Loop

'Initialize
    Cat_num = 1
    Form_num = 1
    Phase = 1
    Call Init_frmFacility(Cat_num, Form_num)

'Show form
    frmFacility.Show

End Sub
```

Wizard Module

```
Sub ModelCreator()

'Save new file
    Call SaveFile
```

```
'Find cell with heading "Facility Cost"
    Worksheets("Construction Cost").Select
    Call FindCell("Facility Cost", 1000, 2, R, C)

'Initialization
    Done = False
    R = R + 1
    Cat_count = 0
    Max_cat_count = 100
    Max_item_count = 100

'Assume sufficient sizes for arrays
    ReDim Item(Max_cat_count, Max_item_count)
    ReDim Qty(Max_cat_count, Max_item_count)
    ReDim Size(Max_cat_count, Max_item_count)
    ReDim Price(Max_cat_count, Max_item_count)
    ReDim Cat(Max_cat_count)
    ReDim Item_count(Max_item_count)

'Store all fields
    Do While Done = False

    'Check two rows down for Category
        If Left(Cells(R + 2, C), 8) = "Category" Then
            'Move to row
                R = R + 2
            'Count categories
                Cat_count = Cat_count + 1
            'Store category name
                Cat(Cat_count) = Right(Cells(R, C), Len(Cells(R, C)) - InStr(Cells(R,
C), "-") - 1)
            'Check next row down for SUBTOTAL
                Do While Left(Cells(R + 1, C), 8) <> "SUBTOTAL"
                    'Move to row
                        R = R + 1
                    'Count items in the category
                        Item_count(Cat_count) = Item_count(Cat_count) + 1
                        Item(Cat_count, Item_count(Cat_count)) = Cells(R,
C).FormulaR1C1
                        Qty(Cat_count, Item_count(Cat_count)) = Cells(R, C +
1).FormulaR1C1
                        Size(Cat_count, Item_count(Cat_count)) = Cells(R, C +
2).FormulaR1C1
                        Price(Cat_count, Item_count(Cat_count)) = Cells(R, C +
3).FormulaR1C1
                Loop
            'Move to row
                R = R + 1
        Else
            'Flag as done
                Done = True
        End If

    Loop

'Initialize
    Cat_num = 1
    Form_num = 1
    Phase = 1
    Call Init_frmFacility(Cat_num, Form_num)

'Show form
    frmFacility.Show
```

```
End Sub
```

Source Code for the form “frmCreateNew”

```
Private Sub btn_frmCreateNew_Cancel_Click()

'Close form
    Unload Me

End Sub

Private Sub btn_frmCreateNew_OK_Click()

'Close form
    Unload Me

If FrmCreateNew_optCreateNew = True Then
    'Reset option variable
        FrmCreateNew_optCreateNew = False
    'Create new model with assistance
        Call ModelCreator

ElseIf FrmCreateNew_optStartOver = True Then
    'Reset option variable
        FrmCreateNew_optStartOver = False
    'Prompt for double check
        msg = "You have chosen to erase all progress in the current file and start
over from scratch.  "
        msg = msg & "This cannot be undone.  Are you sure?"
        choice = MsgBox(msg, vbYesNo, "Confirm Start Over")
        If choice = 6 Then
            'Erase current model
                MsgBox "Once Greg programs it, this command will wipe out any progress
the user made and reset everything to the defaults."
        Else
            Exit Sub
        End If

Else
    'User did not select an option before clicking OK
        MsgBox "Please choose an option, then click OK."
        frmCreateNew.Show

End If

End Sub

Private Sub opt_frmCreateNew_CreateNew_Click()

FrmCreateNew_optCreateNew = True
FrmCreateNew_optStartOver = False

End Sub

Private Sub opt_frmCreateNew_StartOver_Click()

FrmCreateNew_optCreateNew = False
FrmCreateNew_optStartOver = True

End Sub
```

Source Code for the form "frmFacility"

```
'Apply button
Private Sub frmFacility_btn_Apply_Click()

    For i = 1 To Cat_count

    'Find the category name cell
    Call FindCell("Category " & i & " - " & Cat(i), 1000, 2, R, C)

    For k = 1 To Item_count(i)
        Cells(R + k, C) = Item(i, k)
        Cells(R + k, C + 1) = Qty(i, k)
        Cells(R + k, C + 2) = Size(i, k)
        Cells(R + k, C + 3) = Price(i, k)
    Next k

    Next i

End Sub



'OK button
    Private Sub frmFacility_btn_OK_Click()
    Unload Me

    For i = 1 To Cat_count

    'Find the category name cell
    Call FindCell("Category " & i & " - " & Cat(i), 1000, 2, R, C)

    For k = 1 To Item_count(i)
        Cells(R + k, C) = Item(i, k)
        Cells(R + k, C + 1) = Qty(i, k)
        Cells(R + k, C + 2) = Size(i, k)
        Cells(R + k, C + 3) = Price(i, k)
    Next k

    Next i

    End Sub


'Cancel button
    Private Sub frmFacility_btn_Cancel_Click()
    Unload Me
    End Sub


'Category navigation buttons
    'First category button
        Private Sub frmFacility_btn_FirstCat_Click()
        Cat_num = 1
        Form_num = 1
        Call Init_frmFacility(Cat_num, Form_num)
        End Sub
    'Previous category button
        Private Sub frmFacility_btn_PrevCat_Click()
        Cat_num = Cat_num - 1
        Form_num = 1
        Call Init_frmFacility(Cat_num, Form_num)
```

```
        End Sub
    'Next category button
        Private Sub frmFacility_btn_NextCat_Click()
        Cat_num = Cat_num + 1
        Form_num = 1
        Call Init_frmFacility(Cat_num, Form_num)
        End Sub
    'Last category button
        Private Sub frmFacility_btn_LastCat_Click()
        Cat_num = Cat_count
        Form_num = 1
        Call Init_frmFacility(Cat_num, Form_num)
        End Sub

'Form navigation buttons
    'First form button
        Private Sub frmFacility_btn_FirstForm_Click()
        Form_num = 1
        Call Init_frmFacility(Cat_num, Form_num)
        End Sub
    'Previous form button
        Private Sub frmFacility_btn_PrevForm_Click()
        Form_num = Form_num - 1
        Call Init_frmFacility(Cat_num, Form_num)
        End Sub
    'Next form button
        Private Sub frmFacility_btn_NextForm_Click()
        Form_num = Form_num + 1
        Call Init_frmFacility(Cat_num, Form_num)
        End Sub
    'Last form button
        Private Sub frmFacility_btn_LastForm_Click()
        Form_num = Int(Item_count(Cat_num) / 10) + 1
        Call Init_frmFacility(Cat_num, Form_num)
        End Sub
'Clear line item buttons
    Private Sub frmFacility_btn_Clear01_Click()
    frmFacility_txt_Name01.Value = ""
    frmFacility_txt_Qty01.Value = ""
    frmFacility_txt_Size01.Value = ""
    frmFacility_txt_Price01.Value = ""
    End Sub
    Private Sub frmFacility_btn_Clear02_Click()
    frmFacility_txt_Name02.Value = ""
    frmFacility_txt_Qty02.Value = ""
    frmFacility_txt_Size02.Value = ""
    frmFacility_txt_Price02.Value = ""
    End Sub
    Private Sub frmFacility_btn_Clear03_Click()
    frmFacility_txt_Name03.Value = ""
    frmFacility_txt_Qty03.Value = ""
    frmFacility_txt_Size03.Value = ""
    frmFacility_txt_Price03.Value = ""
    End Sub
    Private Sub frmFacility_btn_Clear04_Click()
    frmFacility_txt_Name04.Value = ""
    frmFacility_txt_Qty04.Value = ""
    frmFacility_txt_Size04.Value = ""
    frmFacility_txt_Price04.Value = ""
    End Sub
    Private Sub frmFacility_btn_Clear05_Click()
    frmFacility_txt_Name05.Value = ""
    frmFacility_txt_Qty05.Value = ""
```

```
    frmFacility_txt_Size05.Value = ""
    frmFacility_txt_Price05.Value = ""
    End Sub
    Private Sub frmFacility_btn_Clear06_Click()
    frmFacility_txt_Name06.Value = ""
    frmFacility_txt_Qty06.Value = ""
    frmFacility_txt_Size06.Value = ""
    frmFacility_txt_Price06.Value = ""
    End Sub
    Private Sub frmFacility_btn_Clear07_Click()
    frmFacility_txt_Name07.Value = ""
    frmFacility_txt_Qty07.Value = ""
    frmFacility_txt_Size07.Value = ""
    frmFacility_txt_Price07.Value = ""
    End Sub
    Private Sub frmFacility_btn_Clear08_Click()
    frmFacility_txt_Name08.Value = ""
    frmFacility_txt_Qty08.Value = ""
    frmFacility_txt_Size08.Value = ""
    frmFacility_txt_Price08.Value = ""
    End Sub
    Private Sub frmFacility_btn_Clear09_Click()
    frmFacility_txt_Name09.Value = ""
    frmFacility_txt_Qty09.Value = ""
    frmFacility_txt_Size09.Value = ""
    frmFacility_txt_Price09.Value = ""
    End Sub
    Private Sub frmFacility_btn_Clear10_Click()
    frmFacility_txt_Name10.Value = ""
    frmFacility_txt_Qty10.Value = ""
    frmFacility_txt_Size10.Value = ""
    frmFacility_txt_Price10.Value = ""
    End Sub
```

```
'Private subroutines for changing line item names
    Private Sub frmFacility_txt_Name01_Change()
    Item(Cat_num, frmFacility.frmFacility_lbl_Number01.Caption) =
frmFacility_txt_Name01.Value
    If frmFacility_txt_Name01.Value <> "" Then
        frmFacility_btn_Clear01.Enabled = True
    ElseIf frmFacility_txt_Qty01.Value = "" And frmFacility_txt_Size01.Value = "" And
frmFacility_txt_Price01.Value = "" Then
        frmFacility_btn_Clear01.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Name02_Change()
    Item(Cat_num, frmFacility.frmFacility_lbl_Number02.Caption) =
frmFacility_txt_Name02.Value
    If frmFacility_txt_Name02.Value <> "" Then
        frmFacility_btn_Clear02.Enabled = True
    ElseIf frmFacility_txt_Qty02.Value = "" And frmFacility_txt_Size02.Value = "" And
frmFacility_txt_Price02.Value = "" Then
        frmFacility_btn_Clear01.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Name03_Change()
    Item(Cat_num, frmFacility.frmFacility_lbl_Number03.Caption) =
frmFacility_txt_Name03.Value
```

```
    If frmFacility_txt_Name03.Value <> "" Then
        frmFacility_btn_Clear03.Enabled = True
    ElseIf frmFacility_txt_Qty03.Value = "" And frmFacility_txt_Size03.Value = "" And
frmFacility_txt_Price03.Value = "" Then
        frmFacility_btn_Clear03.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Name04_Change()
    Item(Cat_num, frmFacility.frmFacility_lbl_Number04.Caption) =
frmFacility_txt_Name04.Value
    If frmFacility_txt_Name04.Value <> "" Then
        frmFacility_btn_Clear04.Enabled = True
    ElseIf frmFacility_txt_Qty04.Value = "" And frmFacility_txt_Size04.Value = "" And
frmFacility_txt_Price04.Value = "" Then
        frmFacility_btn_Clear04.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Name05_Change()
    Item(Cat_num, frmFacility.frmFacility_lbl_Number05.Caption) =
frmFacility_txt_Name05.Value
    If frmFacility_txt_Name05.Value <> "" Then
        frmFacility_btn_Clear05.Enabled = True
    ElseIf frmFacility_txt_Qty05.Value = "" And frmFacility_txt_Size05.Value = "" And
frmFacility_txt_Price05.Value = "" Then
        frmFacility_btn_Clear05.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Name06_Change()
    Item(Cat_num, frmFacility.frmFacility_lbl_Number06.Caption) =
frmFacility_txt_Name06.Value
    If frmFacility_txt_Name06.Value <> "" Then
        frmFacility_btn_Clear06.Enabled = True
    ElseIf frmFacility_txt_Qty06.Value = "" And frmFacility_txt_Size06.Value = "" And
frmFacility_txt_Price06.Value = "" Then
        frmFacility_btn_Clear06.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Name07_Change()
    Item(Cat_num, frmFacility.frmFacility_lbl_Number07.Caption) =
frmFacility_txt_Name07.Value
    If frmFacility_txt_Name07.Value <> "" Then
        frmFacility_btn_Clear07.Enabled = True
    ElseIf frmFacility_txt_Qty07.Value = "" And frmFacility_txt_Size07.Value = "" And
frmFacility_txt_Price07.Value = "" Then
        frmFacility_btn_Clear07.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Name08_Change()
    Item(Cat_num, frmFacility.frmFacility_lbl_Number08.Caption) =
frmFacility_txt_Name08.Value
    If frmFacility_txt_Name08.Value <> "" Then
        frmFacility_btn_Clear08.Enabled = True
    ElseIf frmFacility_txt_Qty08.Value = "" And frmFacility_txt_Size08.Value = "" And
frmFacility_txt_Price08.Value = "" Then
        frmFacility_btn_Clear08.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Name09_Change()
    Item(Cat_num, frmFacility.frmFacility_lbl_Number09.Caption) =
frmFacility_txt_Name09.Value
    If frmFacility_txt_Name09.Value <> "" Then
        frmFacility_btn_Clear09.Enabled = True
```

```
    ElseIf frmFacility_txt_Qty09.Value = "" And frmFacility_txt_Size09.Value = "" And
frmFacility_txt_Price09.Value = "" Then
        frmFacility_btn_Clear09.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Name10_Change()
    Item(Cat_num, frmFacility.frmFacility_lbl_Number10.Caption) =
frmFacility_txt_Name10.Value
    If frmFacility_txt_Name10.Value <> "" Then
        frmFacility_btn_Clear10.Enabled = True
    ElseIf frmFacility_txt_Qty10.Value = "" And frmFacility_txt_Size10.Value = "" And
frmFacility_txt_Price10.Value = "" Then
        frmFacility_btn_Clear10.Enabled = False
    End If
    End Sub



'Private subroutines for changing quantities
    Private Sub frmFacility_txt_Qty01_Change()
    Qty(Cat_num, frmFacility.frmFacility_lbl_Number01.Caption) =
frmFacility_txt_Qty01.Value
    If frmFacility_txt_Qty01.Value <> "" Then
        frmFacility_btn_Clear01.Enabled = True
    ElseIf frmFacility_txt_Name01.Value = "" And frmFacility_txt_Size01.Value = "" And
frmFacility_txt_Price01.Value = "" Then
        frmFacility_btn_Clear01.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Qty02_Change()
    Qty(Cat_num, frmFacility.frmFacility_lbl_Number02.Caption) =
frmFacility_txt_Qty02.Value
    If frmFacility_txt_Qty02.Value <> "" Then
        frmFacility_btn_Clear02.Enabled = True
    ElseIf frmFacility_txt_Name02.Value = "" And frmFacility_txt_Size02.Value = "" And
frmFacility_txt_Price02.Value = "" Then
        frmFacility_btn_Clear02.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Qty03_Change()
    Qty(Cat_num, frmFacility.frmFacility_lbl_Number03.Caption) =
frmFacility_txt_Qty03.Value
    If frmFacility_txt_Qty03.Value <> "" Then
        frmFacility_btn_Clear03.Enabled = True
    ElseIf frmFacility_txt_Name03.Value = "" And frmFacility_txt_Size03.Value = "" And
frmFacility_txt_Price03.Value = "" Then
        frmFacility_btn_Clear03.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Qty04_Change()
    Qty(Cat_num, frmFacility.frmFacility_lbl_Number04.Caption) =
frmFacility_txt_Qty04.Value
    If frmFacility_txt_Qty04.Value <> "" Then
        frmFacility_btn_Clear04.Enabled = True
    ElseIf frmFacility_txt_Name04.Value = "" And frmFacility_txt_Size04.Value = "" And
frmFacility_txt_Price04.Value = "" Then
        frmFacility_btn_Clear04.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Qty05_Change()
    Qty(Cat_num, frmFacility.frmFacility_lbl_Number05.Caption) =
frmFacility_txt_Qty05.Value
    If frmFacility_txt_Qty05.Value <> "" Then
```

```
        frmFacility_btn_Clear05.Enabled = True
    ElseIf frmFacility_txt_Name05.Value = "" And frmFacility_txt_Size05.Value = "" And
frmFacility_txt_Price05.Value = "" Then
        frmFacility_btn_Clear05.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Qty06_Change()
    Qty(Cat_num, frmFacility.frmFacility_lbl_Number06.Caption) =
frmFacility_txt_Qty06.Value
    If frmFacility_txt_Qty06.Value <> "" Then
        frmFacility_btn_Clear06.Enabled = True
    ElseIf frmFacility_txt_Name06.Value = "" And frmFacility_txt_Size06.Value = "" And
frmFacility_txt_Price06.Value = "" Then
        frmFacility_btn_Clear06.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Qty07_Change()
    Qty(Cat_num, frmFacility.frmFacility_lbl_Number07.Caption) =
frmFacility_txt_Qty07.Value
    If frmFacility_txt_Qty07.Value <> "" Then
        frmFacility_btn_Clear07.Enabled = True
    ElseIf frmFacility_txt_Name07.Value = "" And frmFacility_txt_Size07.Value = "" And
frmFacility_txt_Price07.Value = "" Then
        frmFacility_btn_Clear07.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Qty08_Change()
    Qty(Cat_num, frmFacility.frmFacility_lbl_Number08.Caption) =
frmFacility_txt_Qty08.Value
    If frmFacility_txt_Qty08.Value <> "" Then
        frmFacility_btn_Clear08.Enabled = True
    ElseIf frmFacility_txt_Name08.Value = "" And frmFacility_txt_Size08.Value = "" And
frmFacility_txt_Price08.Value = "" Then
        frmFacility_btn_Clear08.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Qty09_Change()
    Qty(Cat_num, frmFacility.frmFacility_lbl_Number09.Caption) =
frmFacility_txt_Qty09.Value
    If frmFacility_txt_Qty09.Value <> "" Then
        frmFacility_btn_Clear09.Enabled = True
    ElseIf frmFacility_txt_Name09.Value = "" And frmFacility_txt_Size09.Value = "" And
frmFacility_txt_Price09.Value = "" Then
        frmFacility_btn_Clear09.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Qty10_Change()
    Qty(Cat_num, frmFacility.frmFacility_lbl_Number10.Caption) =
frmFacility_txt_Qty10.Value
    If frmFacility_txt_Qty10.Value <> "" Then
        frmFacility_btn_Clear10.Enabled = True
    ElseIf frmFacility_txt_Name10.Value = "" And frmFacility_txt_Size10.Value = "" And
frmFacility_txt_Price10.Value = "" Then
        frmFacility_btn_Clear10.Enabled = False
    End If
    End Sub


'Private subroutines for changing sizes
    Private Sub frmFacility_txt_Size01_Change()
    Size(Cat_num, frmFacility.frmFacility_lbl_Number01.Caption) =
frmFacility_txt_Size01.Value
    If frmFacility_txt_Size01.Value <> "" Then
```

```
        frmFacility_btn_Clear01.Enabled = True
    ElseIf frmFacility_txt_Name01.Value = "" And frmFacility_txt_Qty01.Value = "" And
frmFacility_txt_Price01.Value = "" Then
        frmFacility_btn_Clear01.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Size02_Change()
    Size(Cat_num, frmFacility.frmFacility_lbl_Number02.Caption) =
frmFacility_txt_Size02.Value
    If frmFacility_txt_Size02.Value <> "" Then
        frmFacility_btn_Clear02.Enabled = True
    ElseIf frmFacility_txt_Name02.Value = "" And frmFacility_txt_Qty02.Value = "" And
frmFacility_txt_Price02.Value = "" Then
        frmFacility_btn_Clear02.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Size03_Change()
    Size(Cat_num, frmFacility.frmFacility_lbl_Number03.Caption) =
frmFacility_txt_Size03.Value
    If frmFacility_txt_Size03.Value <> "" Then
        frmFacility_btn_Clear03.Enabled = True
    ElseIf frmFacility_txt_Name03.Value = "" And frmFacility_txt_Qty03.Value = "" And
frmFacility_txt_Price03.Value = "" Then
        frmFacility_btn_Clear03.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Size04_Change()
    Size(Cat_num, frmFacility.frmFacility_lbl_Number04.Caption) =
frmFacility_txt_Size04.Value
    If frmFacility_txt_Size04.Value <> "" Then
        frmFacility_btn_Clear04.Enabled = True
    ElseIf frmFacility_txt_Name04.Value = "" And frmFacility_txt_Qty04.Value = "" And
frmFacility_txt_Price04.Value = "" Then
        frmFacility_btn_Clear04.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Size05_Change()
    Size(Cat_num, frmFacility.frmFacility_lbl_Number05.Caption) =
frmFacility_txt_Size05.Value
    If frmFacility_txt_Size05.Value <> "" Then
        frmFacility_btn_Clear05.Enabled = True
    ElseIf frmFacility_txt_Name05.Value = "" And frmFacility_txt_Qty05.Value = "" And
frmFacility_txt_Price05.Value = "" Then
        frmFacility_btn_Clear05.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Size06_Change()
    Size(Cat_num, frmFacility.frmFacility_lbl_Number06.Caption) =
frmFacility_txt_Size06.Value
    If frmFacility_txt_Size06.Value <> "" Then
        frmFacility_btn_Clear06.Enabled = True
    ElseIf frmFacility_txt_Name06.Value = "" And frmFacility_txt_Qty06.Value = "" And
frmFacility_txt_Price06.Value = "" Then
        frmFacility_btn_Clear06.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Size07_Change()
    Size(Cat_num, frmFacility.frmFacility_lbl_Number07.Caption) =
frmFacility_txt_Size07.Value
    If frmFacility_txt_Size07.Value <> "" Then
        frmFacility_btn_Clear07.Enabled = True
    ElseIf frmFacility_txt_Name07.Value = "" And frmFacility_txt_Qty07.Value = "" And
frmFacility_txt_Price07.Value = "" Then
```

```
        frmFacility_btn_Clear07.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Size08_Change()
    Size(Cat_num, frmFacility.frmFacility_lbl_Number08.Caption) =
frmFacility_txt_Size08.Value
    If frmFacility_txt_Size08.Value <> "" Then
        frmFacility_btn_Clear08.Enabled = True
    ElseIf frmFacility_txt_Name08.Value = "" And frmFacility_txt_Qty08.Value = "" And
frmFacility_txt_Price08.Value = "" Then
        frmFacility_btn_Clear08.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Size09_Change()
    Size(Cat_num, frmFacility.frmFacility_lbl_Number09.Caption) =
frmFacility_txt_Size09.Value
    If frmFacility_txt_Size09.Value <> "" Then
        frmFacility_btn_Clear09.Enabled = True
    ElseIf frmFacility_txt_Name09.Value = "" And frmFacility_txt_Qty09.Value = "" And
frmFacility_txt_Price09.Value = "" Then
        frmFacility_btn_Clear09.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Size10_Change()
    Size(Cat_num, frmFacility.frmFacility_lbl_Number10.Caption) =
frmFacility_txt_Size10.Value
    If frmFacility_txt_Size10.Value <> "" Then
        frmFacility_btn_Clear10.Enabled = True
    ElseIf frmFacility_txt_Name10.Value = "" And frmFacility_txt_Qty10.Value = "" And
frmFacility_txt_Price10.Value = "" Then
        frmFacility_btn_Clear10.Enabled = False
    End If
    End Sub




'Private subroutines for changing prices
    Private Sub frmFacility_txt_Price01_Change()
    Price(Cat_num, frmFacility.frmFacility_lbl_Number01.Caption) =
frmFacility_txt_Price01.Value
    If frmFacility_txt_Price01.Value <> "" Then
        frmFacility_btn_Clear01.Enabled = True
    ElseIf frmFacility_txt_Name01.Value = "" And frmFacility_txt_Qty01.Value = "" And
frmFacility_txt_Size01.Value = "" Then
        frmFacility_btn_Clear01.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Price02_Change()
    Price(Cat_num, frmFacility.frmFacility_lbl_Number02.Caption) =
frmFacility_txt_Price02.Value
    If frmFacility_txt_Price02.Value <> "" Then
        frmFacility_btn_Clear02.Enabled = True
    ElseIf frmFacility_txt_Name02.Value = "" And frmFacility_txt_Qty02.Value = "" And
frmFacility_txt_Size02.Value = "" Then
        frmFacility_btn_Clear02.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Price03_Change()
    Price(Cat_num, frmFacility.frmFacility_lbl_Number03.Caption) =
frmFacility_txt_Price03.Value
    If frmFacility_txt_Price03.Value <> "" Then
```

```
        frmFacility_btn_Clear03.Enabled = True
    ElseIf frmFacility_txt_Name03.Value = "" And frmFacility_txt_Qty03.Value = "" And
frmFacility_txt_Size03.Value = "" Then
        frmFacility_btn_Clear03.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Price04_Change()
    Price(Cat_num, frmFacility.frmFacility_lbl_Number04.Caption) =
frmFacility_txt_Price04.Value
    If frmFacility_txt_Price04.Value <> "" Then
        frmFacility_btn_Clear04.Enabled = True
    ElseIf frmFacility_txt_Name04.Value = "" And frmFacility_txt_Qty04.Value = "" And
frmFacility_txt_Size04.Value = "" Then
        frmFacility_btn_Clear04.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Price05_Change()
    Price(Cat_num, frmFacility.frmFacility_lbl_Number05.Caption) =
frmFacility_txt_Price05.Value
    If frmFacility_txt_Price05.Value <> "" Then
        frmFacility_btn_Clear05.Enabled = True
    ElseIf frmFacility_txt_Name05.Value = "" And frmFacility_txt_Qty05.Value = "" And
frmFacility_txt_Size05.Value = "" Then
        frmFacility_btn_Clear05.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Price06_Change()
    Price(Cat_num, frmFacility.frmFacility_lbl_Number06.Caption) =
frmFacility_txt_Price06.Value
    If frmFacility_txt_Price06.Value <> "" Then
        frmFacility_btn_Clear06.Enabled = True
    ElseIf frmFacility_txt_Name06.Value = "" And frmFacility_txt_Qty06.Value = "" And
frmFacility_txt_Size06.Value = "" Then
        frmFacility_btn_Clear06.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Price07_Change()
    Price(Cat_num, frmFacility.frmFacility_lbl_Number07.Caption) =
frmFacility_txt_Price07.Value
    If frmFacility_txt_Price07.Value <> "" Then
        frmFacility_btn_Clear07.Enabled = True
    ElseIf frmFacility_txt_Name07.Value = "" And frmFacility_txt_Qty07.Value = "" And
frmFacility_txt_Size07.Value = "" Then
        frmFacility_btn_Clear07.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Price08_Change()
    Price(Cat_num, frmFacility.frmFacility_lbl_Number08.Caption) =
frmFacility_txt_Price08.Value
    If frmFacility_txt_Price08.Value <> "" Then
        frmFacility_btn_Clear08.Enabled = True
    ElseIf frmFacility_txt_Name08.Value = "" And frmFacility_txt_Qty08.Value = "" And
frmFacility_txt_Size08.Value = "" Then
        frmFacility_btn_Clear08.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Price09_Change()
    Price(Cat_num, frmFacility.frmFacility_lbl_Number09.Caption) =
frmFacility_txt_Price09.Value
    If frmFacility_txt_Price09.Value <> "" Then
        frmFacility_btn_Clear09.Enabled = True
    ElseIf frmFacility_txt_Name09.Value = "" And frmFacility_txt_Qty09.Value = "" And
frmFacility_txt_Size09.Value = "" Then
```

```
        frmFacility_btn_Clear09.Enabled = False
    End If
    End Sub
    Private Sub frmFacility_txt_Price10_Change()
    Price(Cat_num, frmFacility.frmFacility_lbl_Number10.Caption) =
frmFacility_txt_Price10.Value
    If frmFacility_txt_Price10.Value <> "" Then
        frmFacility_btn_Clear10.Enabled = True
    ElseIf frmFacility_txt_Name10.Value = "" And frmFacility_txt_Qty10.Value = "" And
frmFacility_txt_Size10.Value = "" Then
        frmFacility_btn_Clear10.Enabled = False
    End If
    End Sub
```

Source Code for the form "frmMain"

```
Private Sub btn_frmMain_Cancel_Click()

Application.ActiveWorkbook.Close

End Sub

Private Sub btn_frmMain_OK_Click()

'Close form
    Unload Me

If FrmMain_optUseAssistance = True Then
    'Reset option variable
        FrmMain_optUseAssistance = False
    'Call ModelCreator subroutine
        Call ModelCreator

ElseIf FrmMain_optByMyself = True Then
    'Reset option variable
        FrmMain_optByMyself = False
Else
    'User did not click an option button
        MsgBox "Please choose an option, then click OK."
        frmMain.Show
End If

End Sub

Private Sub opt_frmMain_UseAssistance_Click()

FrmMain_optUseAssistance = True
FrmMain_optByMyself = False

End Sub

Private Sub opt_frmMain_ByMyself_Click()

FrmMain_optUseAssistance = False
FrmMain_optByMyself = True

End Sub
```

Source Code for ThisWorkbook

```
Private Sub Workbook_Open()
```

```
'Reset main form options
FrmMain_optByMyself = False
FrmMain_optUseAssistance = False

'Show WCCJC toolbar
Application.CommandBars("WCCJC").Visible = True

'Show main form
frmMain.Show

End Sub

Private Sub Workbook_BeforeClose(Cancel As Boolean)

'Close WCCJC toolbar
Application.CommandBars("WCCJC").Visible = False

End Sub
```