ORIE 4820: Spreadsheet-Based Modeling and Data Analysis
Project Risk Analysis
Spring 2013

During this lab exercise, you will augment the Liquair-Pro workbook you completed during last week's lab to include a risk assessment tool based on Monte Carlo simulation. This tool will enable the user to assess the contract NPW risk associated with providing a Liquair-Pro product to a ***new customer*** desiring a ***five-year contract*** using the ***LA-1 tank option***. You will then incorporate this simulation functionality into a bisection search algorithm that will enable the user to (1) set a *target probability for Contract NPW loss*, and (2) determine a price-per-KGallons that achieves this loss probability within a specified tolerance level.

The primary purpose of the lab is to give you practice in ***developing risk assessment tools*** that may be used to compare alternatives and to assist decision making. Once completed, you can use this tool as a template to assess the LA-3 and LA-6 tank options, different desired contract lengths, and other factors.

Topics/Tools we will cover:

- Using a combination of Excel and VBA to ***generate columns of normal random values*** and to ***compute the corresponding Contract NPW values***: **norminv**
- ***Dynamically illustrating simulation output*** using the **frequency** array function
- More VBA functionality:
    - ***Passing parameters*** into subroutines
    - Using ***Application.StatusBar*** to display interim results to the user
    - Using logical and looping constructs to ***create a bisection search algorithm***

Background:

Recall that the *New Customer Pricing* worksheet contains an input area for New Customer Information, including the Average Weekly Demand volume and the Annual Growth Rate. Currently, the worksheet computes the Contract NPW for a contract length of $N = 5$ years, assuming that all parameters are known and fixed quantities. There is no explicit incorporation or evaluation of parameter uncertainty. Today, using the *New Customer Risk Analysis* worksheet, we will address this shortcoming for the Average Weekly Demand volume and the Annual Growth Rate input parameters.

Once completed, the *New Customer Risk Analysis* worksheet will enable the user to: (1) treat the Year 1 Demand Volume and the Annual Growth Rate as *random variables* instead of fixed point estimates, (2) assess the *impact of this uncertainty on Contract NPW*, and (3) price the product so that the *likelihood of Contract NPW loss is limited* to a user-specified probability.

Section 1: Developing the Monte Carlo Simulation Tool

The topmost table on the *New Customer Risk Analysis* worksheet displays four key input parameters, three of which are linked to the values entered on the *New Customer Pricing* worksheet. Note that the average weekly demand volume has been converted to its annual equivalent. The fourth parameter, Price per KGallons, can be specified as the user wishes for purposes of simulation. The macro button directly to the right of the price field will set the price to the (deterministic) Contract NPW breakeven price.

On the right-hand side of the table, the user can specify a *standard deviation for the Year 1 Demand Volume* and a *standard deviation for the Annual Demand Growth Rate* to be used in the Monte Carlo simulation. For purposes of executing the simulation, the Year 1 Demand Volume and the Annual Demand Growth Rate are assumed to be *independent, normal random variables*.

To set up and run the Monte Carlo simulation, we will make use of the tables beneath the input area. We will use some built-in Excel functionality as well as three VBA subroutines.

(1) First, use the **norminv** function to populate the Base Row cells C34 and D34 with randomly generated normal values for Factor 1 and Factor 2 using the specified distribution parameters in cells (D9, G9) and (D13, G13), respectively.

Next, press Alt-F11 to open the Visual Basic Editor (VBE). In the left-hand side project window, select Modules within the project hierarchy and double-click on ***Module 1*** to open it. This module contains skeleton code for several VBA subroutines that you will be working on. You will begin with the `Generate_New_Random_Numbers` subroutine.

(2) `Generate_New_Random_Numbers`: This subroutine's purpose is to populate the first two columns of the Simulation Output Table with 1000 randomly generated *static* values for Factor 1 and Factor 2 (i.e., we want these cells to contain *values*, not formulas). Using the existing code in `Generate_New_Random_Numbers` as a starting point, it remains for you to:

   a. Clear the contents of Simulation_Output_Table
   b. Copy the cell Random_Factor_1 to the range Factor_1_Column. Note that this copies the formula with the active **rand** function embedded.
   c. Copy and Paste Special the values of the Factor_1_Column onto itself.
   d. Repeat steps *a*. and *b*. for Factor 2.
   e. Set the Application.CutCopyMode = False after each paste operation.

(3) `Populate_Table_NPW`: Given fixed values in the Factor 1 and Factor 2 columns of the Simulation_Output_Table, this subroutine's purpose is to populate the third column with the corresponding Contract NPW values by using the *New Customer Pricing* worksheet as a calculator. Note that the subroutine accepts a price value as a *passed-in parameter* from the calling subroutine. (Two different subroutines will eventually call this one.) Using the existing code in `Populate_Table_NPW` as a starting point, it remains for you to:

   a. Set the price cell on the *New Customer Pricing* worksheet to be the passed-in price parameter value.
   b. Cycle through the rows of the table using a For… Next Loop.

c. For each table row, populate the two appropriate cells on the *New Customer Pricing* worksheet with the corresponding Factor values and then place the resulting Contract NPW value in the third column of the table.

(4) `RunNewSimulation`: This subroutine executes the entire Monte Carlo simulation and has *already been completed* for you. Note that it uses local variables to hold the original input values from the *New Customer Pricing* worksheet and resets these values just before the subroutine ends. In the interim, it calls `Generate_New_Random_Numbers` and `Populate_Table_NPW` (using the Factor 4 price in cell E22). It remains for you to:

a. Create a button on the *New Customer Risk Analysis* worksheet to call `RunNewSimulation`.

Section 2: Dynamically Displaying the Simulation Output

Now that the Monte Carlo simulation area is completed, we can create a frequency histogram to visualize the results using the summary table in columns H-L. The Contract NPW column of the Simulation_Output_Table will serve as the data source. Note that the "Bin Upper Limit" cells in column H are linked to the bin size cell L24, so that the user can change the granularity of the histogram *dynamically*. For obvious reasons, the Bin Upper Limit of 0 is fixed at the center of the histogram. Also, the "Contract NPW Range" cells in column I contain descriptive labels for the histogram buckets and have been populated for you. Note how the **text** and **concatenate** functions have been used here. The **text** function replaces its argument with a text equivalent in the format specified. The formats used here are currency formats with "$", suppressed leading zeroes, and comma separators. Negative values are displayed in parentheses. (See the Excel Help pages on these functions for details.) To ***complete the frequency table and histogram***:

(1) Select *all* of the cells in the "Frequency" column (J) that are adjacent to the label cells in column I (not including the header or the total cell at the bottom). Type "=FREQUENCY($E$37:$E$1036,$H$37:$H$51)" and press ***Ctrl-Shift-Enter***. The selected cells should now be populated. Recall that using ***Ctrl-Shift-Enter*** enters the function as an *array formula*, which indicates that it will ***return an entire set of values*** (in this case, the frequency vector of the number of the Contract NPW values that fall into each bucket).

(2) Note that columns K and L contain, respectively, the bucket percentages and cumulative bucket percentages corresponding to column J. In particular, note that because the "0" bin limit is fixed in our histogram, cell L43 depicts the empirical loss probability (i.e., the percentage of simulation trials with Contract NPW < 0). This probability is important for Liquair-Pro to manage. ***The higher the price per KGallons, the lower the loss probability, and vice-versa.***

(3) Select the data in the "Frequency" column, and from the ribbon, select the default "Column" chart via Insert->Charts->Column.

(4) Right click on the chart and then click on "Select Data…". Select the cells in the "Contract NPW Range" column to be the Horizontal (Category) Axis Labels. Enter other chart display options as you wish, then click "OK".

(5) Finally, populate cells E27 and E28 with the average and standard deviation of the 1000 simulated NPW values, respectively. Be sure to *use conditional logic* so that these cells only display values if the simulation output table is populated. Populate cell E29 with the loss probability from the frequency table, again using conditional logic to display only if the table is populated.

Section 3: Automating the Price Search Algorithm

In the final part of the exercise, ***you will write a VBA search subroutine***, `Find_Target_PPKG`, that *finds* a price-per-Kgallons that corresponds to the user-entered target loss probability (cell I29), within the specified tolerance (cell L29). For instance, *if the user enters a 10% target loss probability and a 2% tolerance, then your search algorithm must find a price that yields a loss probability between 8% and 12%*.

(1) `Find_Target_PPKG`: This subroutine's purpose is to find a price per gallon that corresponds to the user-entered target loss probability (cell I29), within the specified tolerance (cell L29). Using the existing code in `Find_Target_PPKG` as a starting point, it remains for you to:

a. Clear the contents of the Contract NPW column in the simulation output table and the Target_PPKG cell I27.

b. Establish upper and lower bounds on the target price:
   I. Code has already been written to find the loss probability associated with the (deterministic) breakeven price. The property Application.Statusbar is used to display to the user (in the status bar) the price that is currently being tested. You need to *compare the loss probability achieved at the breakeven price with the Target_Loss_Prob* to determine whether the breakeven price is a lower or upper bound on the target price.
   II. If the breakeven price is a *lower bound* on the target price, then you need to *find a way to establish an upper bound* so that the target price is "sandwiched" between known upper and lower bounds. If the breakeven price is an *upper bound* on the target price, then you need to *find a way to establish a lower bound* so that the target price is "sandwiched" between known upper and lower bounds.

c. Once you have determined upper and lower bounds on the target price, use a bisection search to isolate the target price and STOP when you have found a price within the tolerance threshold. A **Do While** loop that calls the `Populate_Table_NPW` subroutine has already been created for you. The loop terminates when the current loss probability is within the tolerance range of Target_Loss_Prob. However, within the loop, it remains for you to *appropriately update the upper or lower bound on the target price according to the current loss probability*.

d. Create a button on the *New Customer Risk Analysis* worksheet that calls `Find_Target_PPKG`.