ORIE 4820: Spreadsheet-Based Modeling and Data Analysis
Random Variable Simulation Lab Exercise
Spring 2013

The primary objectives of this exercise are to give you practice ***modeling probability functions*** and using simulation to ***incorporate uncertainty into evaluation models***. To this end, we will develop worksheets to generate four different types of random values.

The template file for this exercise is ***RV-simulation.xlsx***. Download a copy of the file from the course Blackboard site and *save it on your computer*.

Topics/Tools we will cover:

- Using built-in Excel functions to compute and display ***common probability distributions***: **norm.dist**, **poisson.dist**
- Explicitly computing a discrete probability distribution and its parameters: **sumif**, **sumproduct, sqrt**
- ***Generating random values*** from various probability distributions using the inversion method: **rand, norm.inv**, **norm.s.inv, mmult, offset**, **match**
- Using ***one-way data tables*** to perform what-if analysis and to ***automate simulation trials***
- Developing a ***simple evaluation model*** and ***incorporating uncertainty*** into the model
- ***Evaluating simulation output*** to estimate the range and likelihoods of various outcomes

If you have problems or questions at any point during the session, please raise your hand.

Background:

The workbook ***RV-simulation.xlsx*** contains five worksheets: ***Normal***, ***Poisson***, ***Dice Roll***, ***Bivariate Normal***, and ***BVN Chart Values***. On each of the first four (partially completed) worksheets, we will dynamically plot the probability function and/or the cumulative distribution function of the associated random variable(s), and we will randomly generate values from the associated distributions. The fifth worksheet contains bivariate normal densities that are used only to illustrate the distribution specified on the ***Bivariate Normal*** worksheet.

On the Bivariate Normal sheet, we will demonstrate how to randomly generate *pairs of correlated bivariate normal values*. Using this functionality, we will simulate a vector of 1,000 paired values that represent, respectively, morning and afternoon demand for baguettes at a local bakery. These simulated values will form the basis for a profit evaluation model that can be used to help decide the number of baguettes to bake at the start of each day.

Section 1: Generating Normal and Poisson Random Values

The Normal worksheet depicts an example of a commonly used random variable family for which built-in Excel functions exist. Recall that the Normal distribution is ***continuous*** and has two parameters that determine its center and spread: μ (mean) and σ (std deviation). The Excel function **norm.dist** has been used in columns in C and D to compute the associated ***probability distribution function*** (pdf) $f(x)$ and ***cumulative distribution function*** (cdf) $F(x)$ values.

Recall from your class notes that in order to generate a random value from a particular probability distribution using the ***inversion method***, we compute $F^{-1}(U)$, where $F^{-1}(x)$ denotes the cdf inverse function and U~$U[0,1]$ denotes a randomly generated standard uniform value (i.e., a random number between 0 and 1 where every number has an ***equal chance*** of being chosen). Since the **rand** function in Excel generates U~$U[0,1]$, and since the **norm.inv** function in Excel computes the Normal cdf inverse $F^{-1}(x)$, we simply need to combine the two:

(1) In cell E6, type "=NORM.INV(RAND(),$B$3,$E$3)", and press Enter. This results in a randomly generated $N(\mu,\sigma)$ value, with parameters μ and σ specified in cells B3 and E3, respectively. (Note that the parameter cells have been named.)

(2) If you press the F9-key (i.e., Calculate), or if you update the workbook in any way, this number will change – see note below.

Note: The **rand** function is ***active***, so every time you make an update to the workbook, all cells with this function embedded will be recomputed. Sometimes this can be distracting if you are in the process of developing a spreadsheet model. To avoid this, you can temporarily update your spreadsheet options so that calculation is ***manual*** instead of automatic: From the top-left corner of your screen, select File, and click Options. Under Formulas, Calculation options, select "Manual", and click OK. Once you set this option, you must press the F9-key if you want your spreadsheet to recalculate. When you finish development, you can set the calculation option back to "Automatic". (Alternatively, you can replace the **rand** formulas with hard-coded values, but then you will need to re-enter the formulas later.)

Now move to the Poisson worksheet. This is an example of a commonly used ***discrete*** random variable family having a single parameter (λ). In this case, we can use the Excel function **poisson.dist** to compute the ***probability mass function*** (pmf) and cdf values in columns C and D.

(3) In cell C10, type "=POISSON.DIST($B10,$B$3,FALSE)", and press Enter. Copy this formula down the column. (For convenience, the distribution is truncated at $x = 150$ on this worksheet. However, this may be problematic for values of λ greater than 100, since the pmf may not be sufficiently covered.)

(4) In cell D10, type "=POISSON.DIST($B10,$B$3,TRUE)", and press Enter. (You could also accumulate the entries in column C.) Copy this formula down the column.

(5) Now build a line graph to plot the pmf and cdf of the associated Poisson distribution:

a. Select C9:D40 (the first 31 pmf and cdf values, including the headers), and from the ribbon select Insert->Charts->Line.

b. Select the first Line chart type with markers displayed.

c. Right click on the chart and click on "Select Data". Under the "Horizontal (Category) Axis Labels", click "Edit", enter B10:B40 and press "OK".

d. To plot the cdf on a secondary axis, right click on the cdf data line, and select "Format Data Series".

e. On the "Series Options" tab, select Secondary Axis, and press "Close".

(6) In column E, subtract column C from column D, so that for every value k in column B, the corresponding cell in column E contains the value $F(k - 1)$.

These $F(k - 1)$ values are the ***beginning points*** of the segments that define the Poisson inverse cdf. (Unlike the Normal distribution case, there is no built-in Excel function to compute the Poisson inverse cdf. We will have to build it explicitly.) Recall that the cdf $F(k)$ for a discrete random variable is a ***discontinuous step function***. In the case of a Poisson random variable X, only the nonnegative integers have positive probability. So, for any value $x \in \mathbf{[}0, 1\mathbf{)}$, $x \in [F(k - 1), F(k)\mathbf{)} \Leftrightarrow F^{-1}(x) = k$. (An example of this is illustrated on page 6 for the sum of two dice.) We will use this fact to generate a Poisson random value $k = F^{-1}(U)$, where $U \sim U[0,1]$.

(7) In cell E6, type "=OFFSET($B$9,MATCH(RAND(),$E$10:$E$160,1),0,1,1)", and press Enter. This will result in a randomly generated value $k = F^{-1}(U)$ according to the specified Poisson distribution. Here, the **rand** function is used to generate U, and **offset** is used to extract the value of k from column B such that $k = F^{-1}(U)$. The embedded **match** function finds the appropriate row index (relative to the top of the table) based on the $F(k - 1)$ values in column E.

Note that the third parameter of the **match** function – the *match_type* parameter – is set to 1 here. This specifies that the function should return the index of ***the largest array value that is less than or equal to lookup_value***. If the *match_type* parameter is set to 1, the array to be examined ***must*** be sorted in increasing order, or an error will result. If the *match_type* parameter is set to 0 – indicating that an *exact* match is needed – then the array to be examined may be sorted in any order.

Section 2: Characterizing the Probability Distribution of a Discrete RV

Characterizing the probability distribution of a random variable is a fundamental building block in the development of decision models that incorporate uncertainty. On the worksheet Dice Roll, the random variable we are concerned with is the sum of the roll of two six-sided dice. The values that this random variable can assume depend upon the ***numbers on the sides of each die***, as well as the ***likelihood that each side will land face up*** when rolled. Of course, most dice have sides with the integers 1 through 6 appearing exactly once, and most dice are "fair", meaning that there is an equal likelihood (1/6) that each number will be rolled. Based on the "green cell" user input, the "Roll Outcomes and Probabilities" section of the Dice Roll worksheet enumerates all possible combinations of rolls, as well as the likelihoods of each of these combinations.

Notes on the "Roll Outcomes and Probabilities" section:

The upper left-hand table of the Dice Roll worksheet allows the user to specify (in the green cells) the numbers that appear on the sides of each die to be rolled, along with the probabilities that each side will land face-up. Based on the entered values, the inner entries of the table compute the probabilities associated with the possible combinations of die outcomes. Similarly, the lower left-hand table lists the sum totals corresponding to each possible combination.

Conditional formatting has been used on the three total cells to indicate when the probabilities assigned to the sides of one or both dice do not sum to 100%. ***Data validation*** has been used to limit the side values that can be entered to integers between 1 and 6.

A quick way to ***populate a matrix whose entries are based on corresponding row and column values*** is to make judicious use of absolute and relative referencing. In this way you can just populate one cell (usually the uppermost-left) and copy it to the others. To see this:

(1) Delete the inner entries of the upper left-hand table (E7:J12).

(2) Now populate cell E7 with "=$C7*E$5". Note that there is an ***absolute*** reference for the top row "5" and the left column "C", but a ***relative*** reference on the "E" and the "7". (We are multiplying the probability values together here since the two dice are assumed to be ***independent*** of one another.)

(3) Copy the formula in cell E7 across the first row, then down the columns of the table.

Notes on the "Probability Distribution for a Two-Dice Roll" section:

Based on the probabilities for all possible roll combinations, the pmf of the two-dice roll has been computed in column N using the **sumif** function:

sumif(*crit_range*, *criterion*, *range_to_sum*)

In each row of column N, the *crit_range* is the matrix of sum totals in the lower-left table and the *criterion* is the value of the dice roll in column M. The *range_to_sum* is the matrix of probabilities in the upper-left table. Note that the order of the **sumif** function arguments is different than the arguments for **sumifs**.

Column O computes the cdf for the dice roll. A graph depicting the resulting pmf and cdf has already been created. Note that different chart types have been used for these series.

Column P is simply the difference between column O and column N. As before, we will use these values to generate random dice rolls according to the probability distribution in column N.

Computing the distribution parameters in N4:N6:

(4) Recall that the ***expected value*** of a discrete random variable Y is defined to be:

$$E[Y] = \sum_{k=-\infty}^{\infty} k \cdot P(Y = k).$$

In cell N4, compute the expected value of the dice roll using the **sumproduct** function:

sumproduct(*list_of_values, list_of_probabilities*)

(5) In cell N5, compute the ***variance*** of the dice roll. Note that the variance of a discrete random variable Y can be written as:

$$Var[Y] = \left(\sum_{k=-\infty}^{\infty} k^2 P(Y = k) \right) - E[Y]^2$$

Therefore, you can compute *Var*[*Y*] by using:

sumproduct(*list_of_values*, *list_of_values*, *list_of_probabilities*) - (*E*[*Y*] * *E*[*Y*])

Note that **sumproduct** is being used on ***three*** arrays here to compute the value inside the parentheses in the above variance expression.

(6) In cell N6, compute the ***standard deviation*** of the dice roll using the **sqrt** function.

Note that if the user changes one or more of the die side values, the probability distribution for the dice roll changes, as do the distribution's mean and standard deviation. Suppose that we wanted to capture how these distribution parameters change as a function of changing the "1" side of Die #1 in cell E6. Excel has a built-in ***data table*** tool that greatly simplifies this sort of "what-if" analysis.

(7) A ***one-way data table*** allows the user to specify a set of different "what-if" values for a *single input parameter*. For each input value, it *returns one or more computed quantities that the user specifies*. Data tables are a particularly useful tool for conducting sensitivity analysis. We will see them many times this semester. To use a one-way data table to perform sensitivity analysis on the dice roll probability distribution parameters as a function of the value of the first side of Die #1:

a. Populate cell E27 with "=$N$4". Note that this formula refers to the cell with the computed distribution mean. This is the output value that will populate this column of the data table for each designated input value. Populate cell F27 with "=$N$6" to capture the distribution standard deviation.

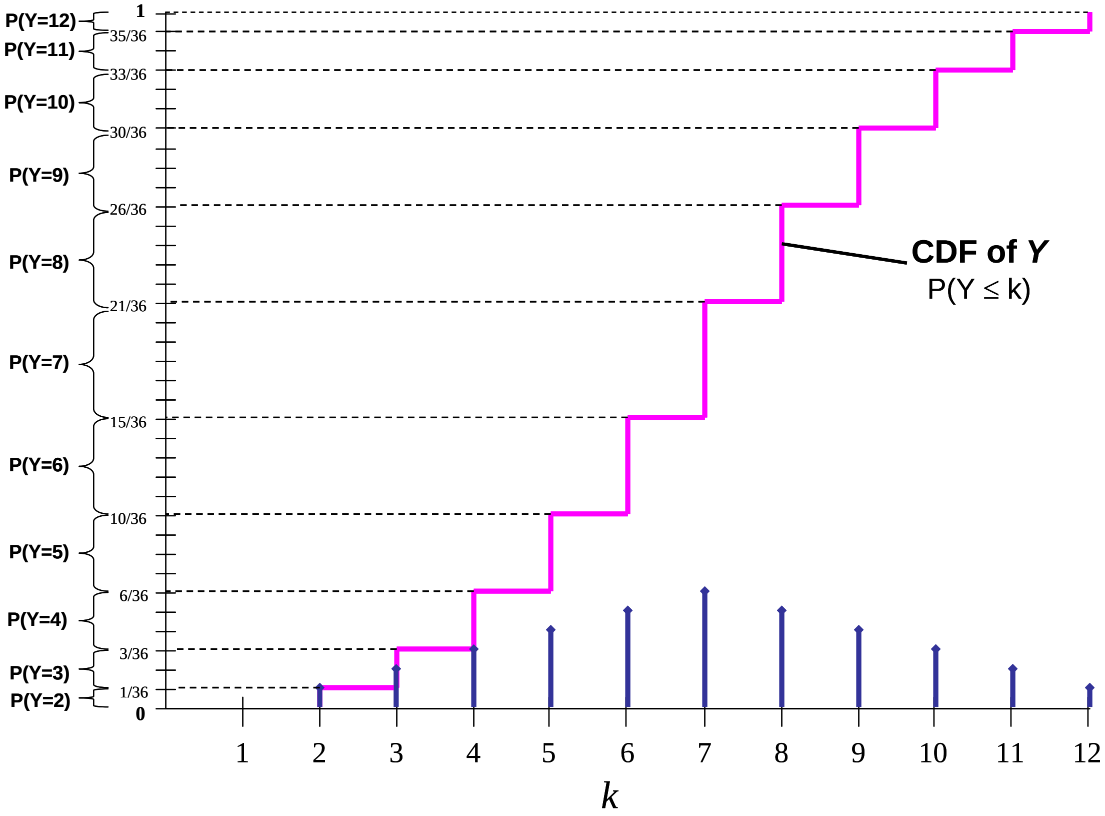
b. Highlight the table range D27:F33 (the rectangle containing the column of input values and the cells you just populated, but not the text headers). From the ribbon, select Data->Data Tools->What-If Analysis->Data Table…

c. Select E6 as the Column Input cell (i.e., each value in the input *column* will effectively be "substituted" into cell E6), leave the Row Input cell blank, and press OK. The table should now be populated with the resulting values.

Section 3: Generating a Vector of Random Values Using a Data Table

Consider the illustration on the next page, which depicts both the probability mass distribution and the cumulative distribution function for a two-dice roll (assuming fair dice).

Note that the segments depicted on the vertical axis ***exactly*** partition the real number interval between 0 and 1. Moreover, the ***heights*** of these segments correspond precisely to the probabilities associated with the dice roll outcomes (2 through 12). Thus, we can ***map*** every real number between 0 and 1 ***uniquely*** to one of the dice roll outcomes. For instance, the number 0.0004 corresponds to the value k = 2. The number 0.5127 corresponds to the value k = 7, and so forth. We have already seen how using **rand** in conjunction with our mapping gives us a way to generate dice roll outcomes according to the correct probability distribution.



The values in column P on the Dice Roll worksheet are precisely the ***beginning points*** of the segments needed to map random numbers to dice rolls. Therefore:

(1) In cell H23, type "=OFFSET($M$9,MATCH(5%,CDF_Mapping_Table,1),0,1,1)", and press Enter. The result should be the number "3", since 5% falls into the segment [2.78%, 8.33%), which is associated with P[Y = 3].

(2) Go back to cell H23 and replace the first parameter of **match** with "RAND()**"**. This will result in a random dice roll according to the associated probability distribution.

(3) We have seen how a one-way data table can be used to conduct sensitivity analysis by varying a single input value. In the same manner, a data table can also be used to conduct simple simulations when the input value to be varied is a randomly generated value (i.e., when the input value simply depends upon a new calculation of the active **rand** function). For example, we can use a one-way data table to ***generate an entire vector of random dice roll values*** as follows:

   a. Populate cell I27 with "=$H$23", the cell with the randomly generated value.

   b. Highlight the table range H27:I37 (the rectangle containing the trial numbers and the I27 cell, but not the text headers) and from the ribbon select Data->Data Tools-> What-If Analysis->Data Table…

   c. Select H27 as the Column Input cell (this is just a dummy substitution – the trial numbers do not need to be "substituted" anywhere, so we simply choose a blank cell), leave the Row Input cell blank, and press OK. The table should now be populated with different randomly generated values.

Section 4: Developing an Evaluation Model

Now move to the Bivariate Normal worksheet. The business scenario we are going to model here is simple to describe. Each day in the early morning, a local bakery produces baguettes for sale that day. Each baguette costs $1.00 to produce. The selling price of baguettes when the bakery opens in the morning is $3.00, but the price drops to $1.50 in the afternoon.

Total baguette demand is independent and identically distributed from day to day. On any given day, however, morning demand (*X*) and afternoon demand (*Y*) have a ***bivariate normal joint probability distribution***, with the five parameters μ_X, σ_X, μ_Y, σ_X, and ρ_{XY} specified in the corresponding user-changeable green cells. The bivariate normal distribution has pdf:

$$f(x,y)=\frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}}e^{-\left\{\frac{1}{2(1-\rho^2)}\left[\frac{(x-\mu_X)^2}{\sigma_X^2}-2\rho\frac{(x-\mu_X)}{\sigma_X}\cdot\frac{(y-\mu_Y)}{\sigma_T}+\frac{(y-\mu_Y)^2}{\sigma_Y^2}\right]\right\}}$$

Any leftover baguettes at the end of the day are donated to a nearby soup kitchen. The bakery wants to know ***how many baguettes to produce each morning to maximize profit***.

Because the profit margin on baguettes is different in the morning than in the afternoon, we need to distinguish morning demand from afternoon demand in our evaluation model. Moreover, the extent to which morning and afternoon demand are correlated (ρ_{XY}) directly impacts the variability of total demand and, consequently, the probability distribution of profit.

Setting aside demand uncertainty for a moment, let us ***first assume that we know what the morning and afternoon demand values are*** on a particular day and develop a framework that computes the total profit for the day:

(1) Enter the value 45 in cell N30 (to represent demand on a given morning) and the value 35 in cell O30 (to represent demand that afternoon). Enter the value 70 in the decision cell P24 (number of baguettes to bake).

(2) In cell P30, sum the entries in N30 and O30.

(3) In cell Q30, determine the total number of baguettes that are sold in the morning (i.e., the *smaller* of the number baked and the morning demand).

(4) In cell R30, determine the total number of baguettes that are sold that afternoon (i.e., the *smaller* of the number leftover from the morning and the afternoon demand).

(5) In cell S30, compute the associated profit. With the default values, this is:

$3*(number sold in morning) + $1.50*(number sold in afternoon) - $1*(number baked)

(6) Change the values in cells N30 and O30 to validate your computations.

Now that we have a framework established for computing profit when demand is known, it remains to incorporate demand uncertainty into the model and evaluate the effect on profit. This can be harder than it sounds. Although we know what the joint probability distribution of the demand vector (*X*, *Y*) looks like (in fact, one can even show that the total demand random variable $W = X + Y$ is normal), it is not immediately clear what the probability distribution of the daily ***profit*** looks like.

In order to determine how the demand uncertainty affects profit for a given supply decision, we can either try to directly compute the probability distribution of profit using analytical methods (hard), or we can approximate the distribution and/or its parameters using simulation (easy). In order to do the latter, we must first be able to ***randomly generate demand vectors according to the specified bivariate normal probability distribution***.

Section 5: Generating Correlated Bivariate Normal Random Values

Although we omit the details here, it is a fortunate fact that one can generate a bivariate normal random vector (X, Y) using the following linear transformation on a random vector of values (N_X, N_Y) that are generated from ***independent standard normal*** distributions:

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} \mu_X \\ \mu_Y \end{bmatrix} + \begin{bmatrix} \sigma_X & 0 \\ \rho\sigma_Y & \sigma_Y\sqrt{1-\rho^2} \end{bmatrix} \begin{bmatrix} N_X \\ N_Y \end{bmatrix}$$

The 2x2 lower triangular matrix (L) in the transformation is, in fact, the Cholesky decomposition of the symmetric variance-covariance matrix Σ_{XY} that satisfies $L\, L^T = \Sigma_{XY}$. This result extends to the multivariate normal case as well. We will accomplish this transformation on our worksheet as follows:

(1) Enter the appropriate formulas for the **μ** vector and L matrix in cells M9:M10 and O9:P10, respectively.

(2) In cells R9 and R10, generate $N(0,1)$ values by entering "=NORM.S.INV(RAND())".

(3) In cells O12:O13, compute the linear transformation using the **mmult *array function***: With cells O12:O13 selected, type "=M9:M10+MMULT(O9:P10,R9:R10)" and press ***Ctrl-Shift-Enter***. The function **mmult** performs standard matrix multiplication on the entered arguments. Remember that *order matters* in matrix multiplication.

(4) Cells O12:O13 should now display a pair of values generated from the specified bivariate normal distribution. Pressing the F9-key will give you new values in these cells.

Section 6: Incorporating Uncertainty Using Simulation

Now that we have a mechanism for generating (X,Y) pairs of demand values, we can incorporate demand uncertainty into our evaluation model using simulation:

(1) Again, we will use a ***one-way data table*** to generate 1,000 pairs of random (X,Y) values. In this case, we will have one (dummy) input parameter column and two output columns:

   a. Populate cell N29 with "=MAX(ROUND(O12,0),0)". This will round the generated morning demand values to the nearest non-negative integer.

   b. Populate cell O29 with "=MAX(ROUND(O13,0),0)". This will round the generated afternoon demand values to the nearest non-negative integer.

c. Highlight the table range M29:O1029 (the rectangle containing the trial numbers and the cells you just populated, but not the text headers) and from the ribbon select Data->Data Tools->What-If Analysis->Data Table…

d. Select M29 as the Column Input cell (this is just a dummy substitution here – any inert cell will work), leave the Row Input cell blank, and press OK. The table should now be populated with generated values that have been rounded. Note that a scatterplot of the generated pairs has already been created for you.

(2) Now copy the formulas in cells P30:S30 down their respective columns. This yields (in column S) 1,000 iid samples from the profit distribution.

(3) Compute the following summary statistics:

a. Cell U20: Sum of values in column P.

b. Cell U21: Sum of values in columns Q and R.

c. Cell V21 (off to the side): U21/U20, the % of total demand that was satisfied.

d. Cell U23 is U22-U21. (Cell U22 is the total units produced.)

e. Cell V23 (off to the side): U23/U22, the % of production that was discarded.

f. Cells U25 and U26: Average and std dev, respectively, of the values in column S.

(4) Use the partially completed area in cells U30:V47 to construct a frequency histogram of (simulated) daily profit using the **frequency** function.

Questions to consider:

- ***If the bakery makes 70 baguettes, give a rough range for the average daily profit assuming the parameter values for demand are:*** $\mu_X = 40$, $\sigma_X = 10$, μ_Y,=30, σ_X, = 10 and $\rho_{XY} = 0$?

- ***Consider the frequency table and histogram for profit that you constructed in cells U30:V47. Does profit appear to be normally distributed? What does the histogram look like if the bakery makes 50 baguettes? 40 baguettes? 100 baguettes?***

- ***Using the current cost and revenue parameters, how could you determine the "optimal" number of baguettes to make?***

- ***Set*** $\rho_{XY} = 0.9$ ***and re-simulate. How do the profit statistics change? Can you explain this?***

- ***Set*** $\rho_{XY} = -0.9$ ***and re-simulate. How do the profit statistics change? Can you explain this?***

- ***Set*** $\rho_{XY} = 0$ ***again and instead of $1.50, set the afternoon selling price to $3.00 (same as the morning price). Assuming that demand is unaffected by the price change, how does the "optimal" number of baguettes change?***

- ***Set the afternoon selling price to $1.00. Now how does the "optimal" number of baguettes change?***