APPENDIX 1

@Risk Crib Sheet

@Risk Icons

Once you are familiar with the function of the @Risk icons, you will find @Risk easy to learn. Here is a description of the icons.

Opening an @Risk Simulation

This icon allows you to open up a saved @Risk simulation. I do not recommend saving simulations. Instead, I paste results into a spreadsheet.

Saving an @Risk Simulation

This icon allows you to save an @Risk simulation, including data and simulation settings.

Simulation Settings

This icon allows you to control the settings for the simulation. Clicking on this icon activates the dialog box shown in Figure 1. There follows a description of what each of the tabs can do.

Iterations Tab

Various options are associated with the Iterations tab.

#Iterations #Iterations is how many times you want @Risk to recalculate the spreadsheet. For example, choosing 100 iterations means that 100 values of your output cells will be tabulated.

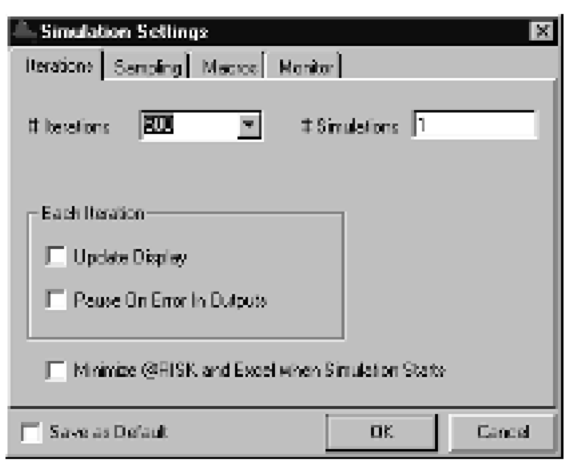


FIGURE 1

#Simulations Leave this at 1 unless you have a =RISKSIMTABLE functon in the spreadsheet. In this case, choose #Simulations to equal the number of values in SIMTABLE. For example, if we have the formula =RISKSIMTABLE({100,150,200,250,300}) in cell A1, set #Simulations to 5. The first simulation will place 100 in A1, the second simulation will place 150 in A1, and the fifth simulation will place 300 in A1. #Iterations will be run for each simulation.

Pause on Error Checking this box causes @Risk to pause if an error occurs in any cell during the simulation. @Risk will highlight the cells where the error occurs.

Update Display Checking this box causes @Risk to show the results of each iteration on the screen. This is nice, but it slows things down.

See Figure 2. The Sampling tab options are as follows.

Sampling Type While a little slower, Latin Hypercube sampling is much more accurate than Monte Carlo sampling. To illustrate, Latin Hypercube guarantees for a given cell that 5% of observations will come from the bottom 5th percentile of the actual random variable, 5% will come from the top 5th percentile of the actual random variable, etc. If we choose Monte Carlo sampling, 8% of our observations may come from the bottom 5% of the actual distribution, when in reality only 5% of observations should do so. When simulating financial derivatives, it is crucial to use Latin Hypercube.

Standard Recalc If you choose Expected Value, you obtain the expected value of the random variable unless the random variable is discrete. Then you obtain the possible value of the random variable that is closest to the random variable's expected value. For instance, for a statement

=RISKDISCRETE({1,2,},{.6,.4})

the expected value is $1(.6) + 2(.4) = 1.4$, so Expected Value enters a 1.

If you choose the Monte Carlo option, *when you hit F9, all the random cells will recalculate. This makes it much easier to understand and debug the spreadsheet.* Thus, with Monte Carlo selected,

=RISKDISCRETE({1,2,},{.6,.4})

will return a 1 60% of the time and a 2 40% of the time.

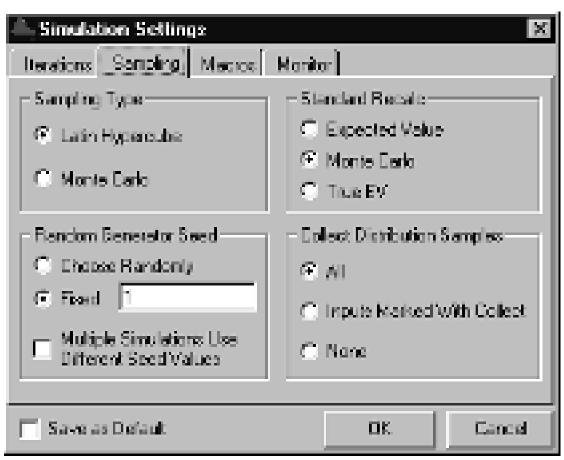


FIGURE 2

If you choose the True EV option, then the actual expected value of the random variable will be returned. Thus,

=RISKDISCRETE({1,2,},{.6,.4})

will yield a 1.4.

Collecting Distribution Samples Check All if you want to get Tornado Graphs, Scenario Analysis, or Extract Data. Also check this box if you want statistics on cells generated by @Risk functions. You can always check this box if you like, but if you have many @Risk functions in your spreadsheet, checking the box will slow down the simulation. Checking Inputs Marked With Collect will collect data on a subset of your risk functions marked with Riskcollect.

Random Number Generator Seed When the seed is set to 0, each time you run a simulation, you will obtain different results. Other possible seed values are integers between 1 and 32,767. Whenever a nonzero seed is chosen, the same values for the input cells and output cells will occur. For example, if we choose a seed value of 10, each time we run the simulation, we will obtain exactly the same results.

Autoconvergence

Under #Iterations, you may select Auto. See Figure 3. You may then select a percentage such as 1%. Then @Risk keeps running until during the last 100 iterations, the mean and standard deviation change by at most 1%. This can be a lot of iterations! I prefer to choose the number of iterations myself by setting

$$\frac{2s}{\sqrt{n}}$$

equal to the desired level of accuracy for the output cell's mean. Here, s = standard deviation of output cell for a trial simulation (say, 400 iterations). For example, if a trial simulation yields $s = 100$ and I want to be 95% sure that I am estimating the population mean within 10, I need

$$\frac{2(100)}{\sqrt{n}} = 10$$

or $n = 400$.

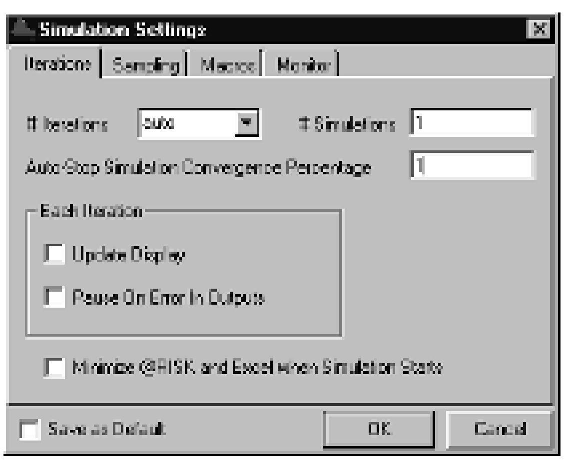


FIGURE 3

Macro Tab

See Figure 4. The Macro tab enables @Risk to run a macro before or after each iteration of a simulation. For example, checking After Each Iteration's Recalc and entering Macro1 after evaluating each ouput cell would result in the following sequence of events:

- Compute @Risk functions and calculate output cells.
- Run Macro1.
- Compute @Risk functions and calculate output cells, etc.

Select Output Cells

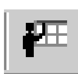
This icon enables you to select an output cell or cells for which @Risk will create statistics. Simply select a range of cells and click on the icon to select the range as output cells. You may select as many ranges as you desire.

List Input and Output Cells

This icon lists all output cells. Also listed are cells containing @Risk functions. These are called input cells. From this list, you can change the names of output cells or delete output cells.

Run Simulation

This icon starts the simulation. The status of the simulation is shown in the lower left-hand corner of your screen. Hitting the Escape key allows you to terminate the simulation.



Show Results

This icon allows you to see results. There are two windows:

- Summary Results, containing Minimum, Mean, and Maximum for all input and output cells.
- Simulation Statistics, containing more detailed statistics.

Clicking the Hide icon will send you back to your worksheet. To paste your statistics into your worksheet, simply select a window and Edit Copy Paste it into the worksheet.

Define Distribution Icon

This icon allows you to see the mass function or density function for any random variable. You may also use this icon directly to enter any @Risk formula into a cell.

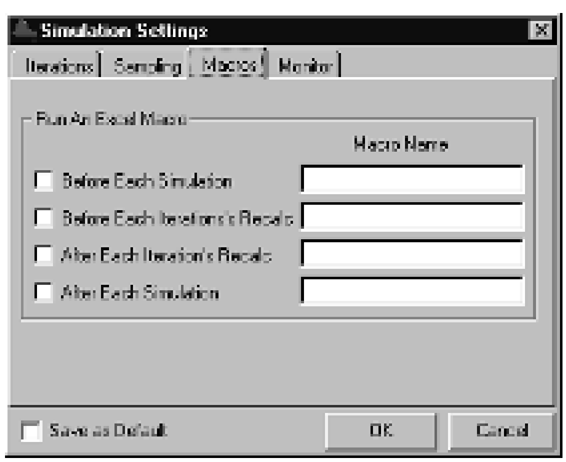


FIGURE 4

Graphing

To obtain a graph, right click on the cell from the Explorer interface. Then choose the type of graph desired. To copy the graph into Excel, right click on the graph and select Copy or Graph in Excel.

A histogram gives the fraction of iterations assuming different values. The histogram in Figure 5 was generated for a cell containing the formula

=RISKNORMAL(100,15)

The histogram indicates that the input cell was bell-shaped and that the most common values of the input cell were around 100.

For a cumulative ascending graph (Figure 6), the y-axis gives the fraction of iterations yielding a value $\leq$ the value on the x-axis. Thus, about 50% of all iterations in this case yielded a value $\leq$ 100.

For a cumulative descending graph, the y-axis gives the fraction of iterations yielding a value $\geq$ the value on the x-axis. In Figure 7, this input cell exceeded 85 about 84% of the time.

An area graph replaces bars with smooth areas. A fitted curve smooths out the variation in bar heights before creating an area graph.

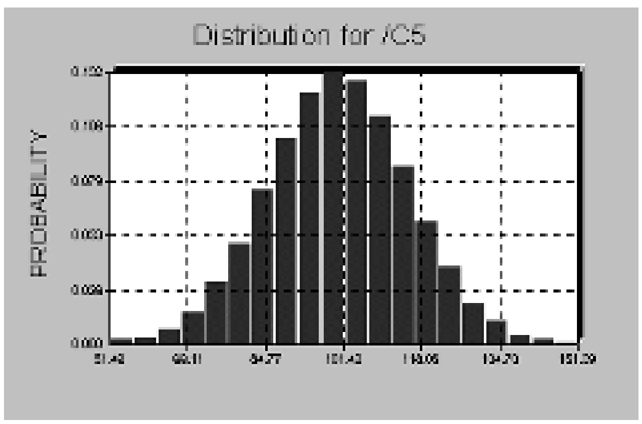


FIGURE 5
Histogram

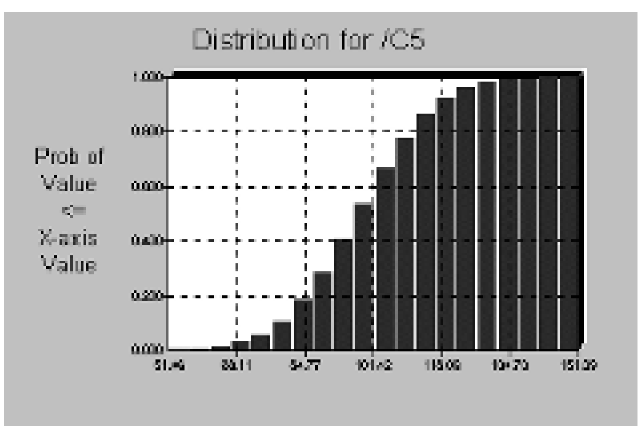


FIGURE 6
Cumulative Ascending Graph

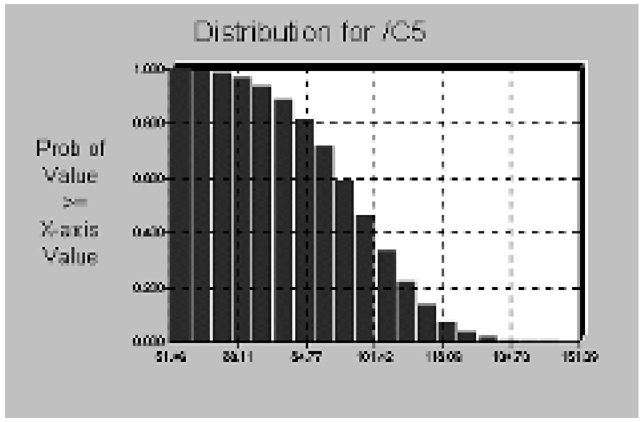


FIGURE 7
Cumulative Descending Graph

Targets

At the bottom of the Simulation Statistics window is a Target option. You may enter a Value or Percentile, and @Risk fills in the one you left out. For

=RISKNORMAL(100,15)

we obtained the following results:

```
Target #1 (Value)=      85
Target #1 (Perc%)=      15.87%
Target #2 (Value)=      130
Target #2 (Perc%)=      97.75%
Target #3 (Value)=      114.9159
Target #3 (Perc%)=      84%
```

- We entered Target#1(Value) of 85, and @Risk reported that the cell was ≤85 15.87% of the time.
- We entered Target#2(Value) of 130, and @Risk reported that the cell was ≤130 97.75% of the time.
- We entered Target#3(Perc%) of 84%, and @Risk reported that 84% of the time, the cell was ≤114.92.

Extracting Data

Sometimes you may want to see the values of @Risk functions and output cells that @Risk created on the iterations run. If so, check Collect Distribution Samples under Simulation Settings and then click on Data in the Results window. You can then Edit Copy Paste the data to your spreadsheet and subject it to further analysis.

Sensitivity

If you want a Tornado Graph, right click on the output cell and select Tornado Graph. This also requires that you check Collect Distribution Samples. You may choose either a Correlation or a Regression graph. Tornado graphs let you know which input cells have the largest influence on your output cell(s).

@Risk Functions

We now illustrate some of the most useful @Risk functions.

The RISKDISCRETE Function

This generates a discrete random variable that takes on a finite number of values with known probabilities. See Figure 8. First, enter the possible values of the random variable

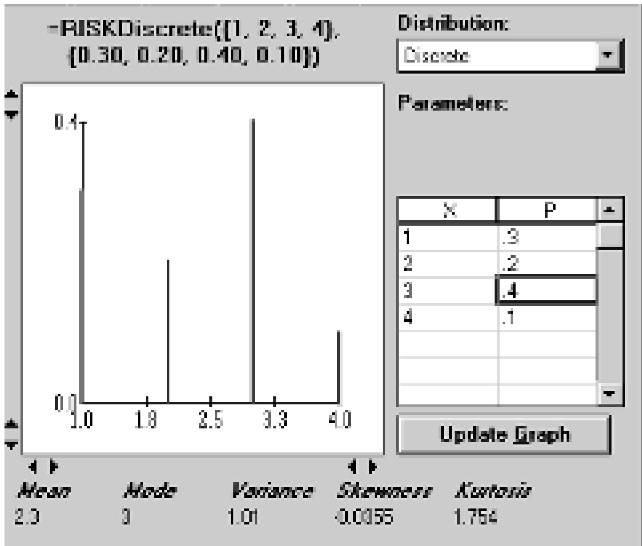


FIGURE 8

and then the probability for each value. Thus, =RISKDISCRETE({1,2,3,4},{.3,.2,.4,.1}) would generate 1 30% of the time, 2 20% of the time, 3 40% of the time, and 4 10% of the time.

If the values and probabilities were entered in A2:B5, we could have entered this random variable with formula

=RISKDISCRETE(A2:A5, B2:B5)

The RISKSIMTABLE Function

Suppose we enter

=RISKSIMTABLE({100,150,200,250,300})

in cell A5, and #Iterations is 100. If we change #Simulations to 5, then on the first simulation, 100 iterations are run with 100 in cell A5. On the second simulation, 100 iterations are run with 150 in cell A5. Finally, on the fifth simulation, 100 iterations are run with 300 in cell A5. If the five arguments for the =RISKSIMTABLE function were in B1:B5, we could have also entered the =RISKSIMTABLE function as

=RISKSIMTABLE(B1:B5)

The RISKDUNIFORM Function

See Figure 9. We use the RISKDUNIFORM function when a random variable assumes several equally likely values. Thus,

=RISKDUNIFORM({1,2,3,4})

is equally likely to generate 1, 2, 3, or 4. If 1, 2, 3, 4 were entered in A1:A4, then we could have entered

=RISKDUNIFORM(A1:A4)

The RISKBINOMIAL Function

See Figure 10. Use the =RISKBINOMIAL function when you have repeated independent trials, each having the same probability of success. For example, if there are 5 competi-

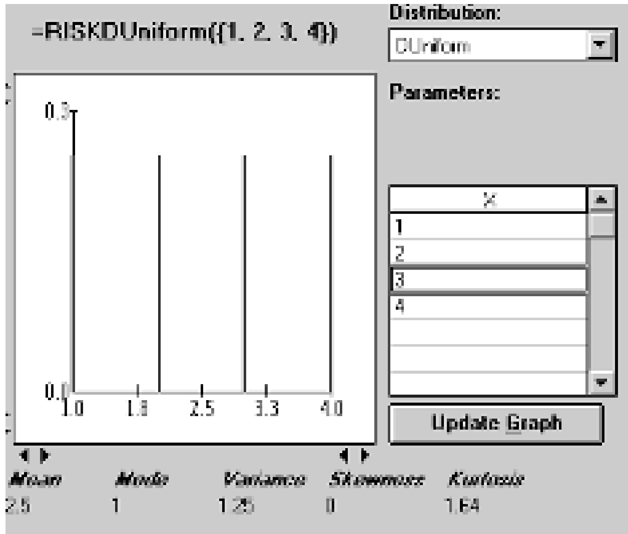


FIGURE 9

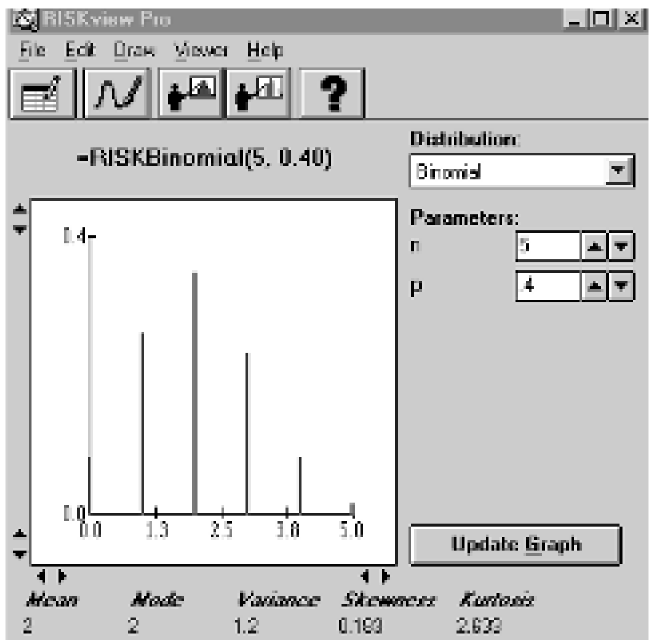


FIGURE 10

tors who might enter an industry this year, each competitor has a 40% chance of entering, and entrants are independent, then we could model this situation with the formula

=RISKBINOMIAL(5,.4)

The RISKNORMAL Function

See Figure 11. Use this function to model a continuous, symmetric (or bell-shaped) random variable. The formula

=RISKNORMAL(100,15)

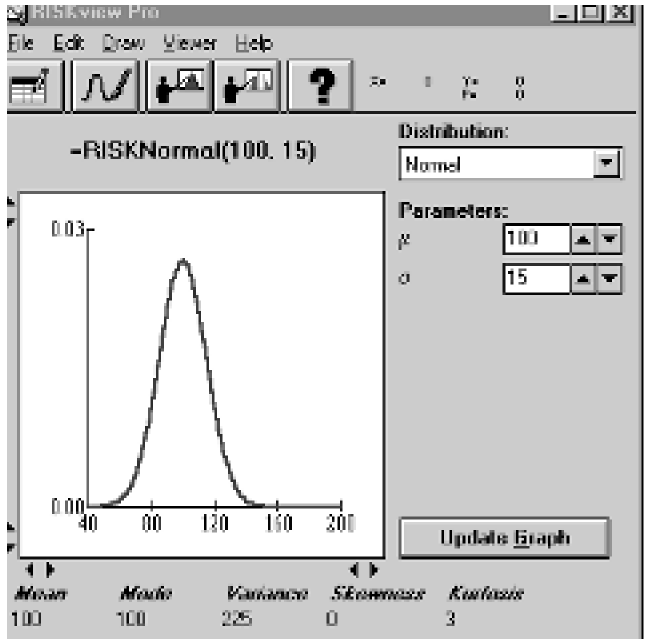


FIGURE 11

will yield

- a value between 85 and 115 68% of the time
- a value between 70 and 130 95% of the time
- a value between 55 and 145 99.7% of the time

The RISKTRIANG Function

See Figure 12. This function enables us to model a nonsymmetrical continuous random variable. It generalizes the well-known idea of best-case, worst-case, and most likely scenarios. For example,

$$=\text{RISKTRIANG}(.2,.4,.8)$$

could be used to model market share if we felt that the worst-case market share was 20%, the most likely market share was 40%, and the best-case market share was 80%. Note that the probability that the market share is between 30% and 40% would be the area under this triangle between .3 and .4. The entire triangle has an area of 1. This fact determines the height of the triangle.

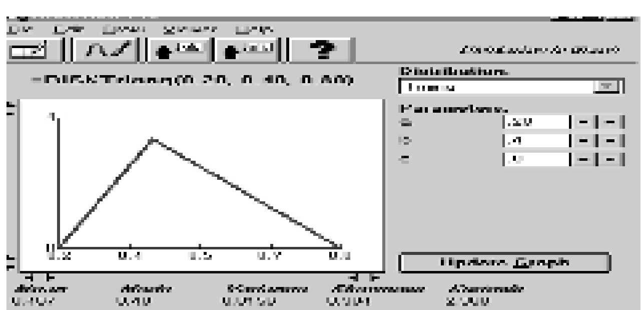


FIGURE 12

The RISKTRIGEN Function

See Figure 13. Sometimes we want to use a triangular random variable, but we are not sure of the absolute best and worst possibilities. We may believe that there is a 10% chance that market share will be less than or equal to 30%, that the most likely share is 40%, and that there is a 10% chance that share will exceed 75%. The RISKTRIGEN function is used in this situation. The formula

=RISKTRIGEN(.3,.4,.75,10,90)

would be appropriate for this situation. Then @Risk draws a triangle that yields

- A 10% chance that market share is less than or equal to 30%. This requires a worst possible market share of around 20%.
- A most likely market share of 40%.
- A 10% chance that market share is greater than or equal to 75%. This requires a best possible market share of around 95%.

Again, the probability of a market share between 20% and 50% is just the area under the triangle between 20% and 50%.

The RISKUNIFORM Function

See Figure 14. Suppose a competitor's bid is equally likely to be anywhere between 10 and 30 thousand dollars. This can be modeled by a uniform random variable with the formula

=RISKUNIFORM(10,30)

Again, this function makes any bid between 10 and 30 thousand dollars equally likely. The probability of a bid between 15 and 28 thousand would be the area of the rectangle bounded by $x = 15$ and $x = 28$. This would equal $(28 - 15)(.05) = .65$.

The RISKGENERAL Function

What if a continuous random variable does not appear to follow a normal or a triangular distribution? We can model it with the =RISKGENERAL function.

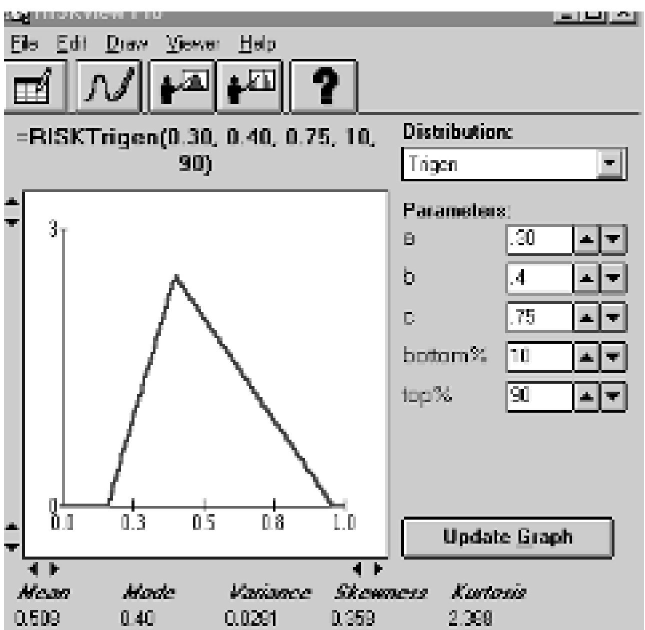


FIGURE 13

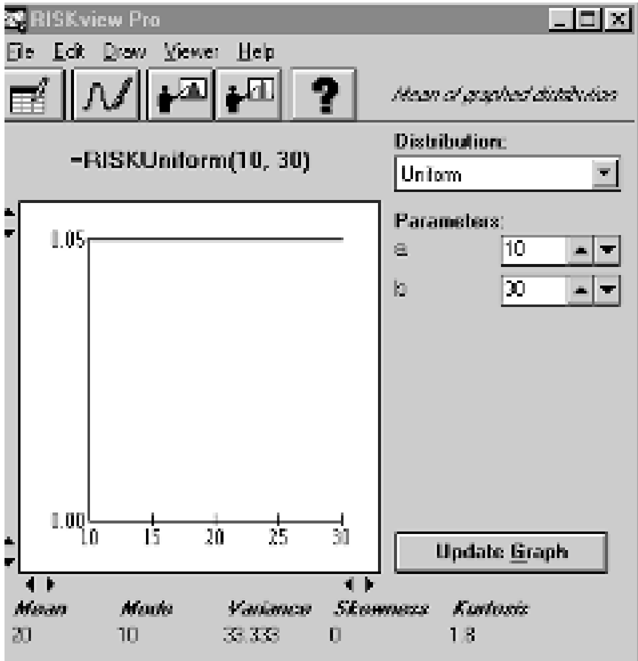


FIGURE 14

Suppose that a market share of between 0 and 60% is possible, and a 45% share is most likely. There are five market-share levels for which we feel comfortable about comparing relative likelihood. (See Table 1.) Thus, a market share of 45% is 8 times as likely as 10%; 20% and 55% are equally likely; etc. Note that this distribution cannot be triangular, because then 20% would be (20/45) as likely as peak of 45%, and 20% would be .75 as likely as 45%. To model this, enter the formula

=RISKGENERAL(0,60,{10,20,45,50,55},{1,6,8,7,6})

The syntax of RISKGENERAL is as follows:

- Begin with the smallest and largest possible values.
- Then enclose in {} the numbers for which you feel you can compare relative likelihoods.
- Finally, enclose in {} the relative likelihoods of the numbers you have previously listed.

Running this in @Risk yields the output shown in Figure 15. Note that 20 is 6/8 likely as 45; 10 is 1/8 as likely as 45; 50 is 7/8 as likely as 45; 55 is 6/8 as likely as 45; etc. In be-

TABLE 1

Market Share	Relative Likelihood
10%	1
20%	6
45%	8
50%	7
55%	6

	A	B	C	D	E	F	G
1		**EXAMPLE OF**					
2		**RISKGENERAL**					
3		**DISTRIBUTION**					
4							
5				Minimum	0		
6				Maximum	60		
7				**Specified Points**			
8				10	1		
9				20	6		
10				45	8		
11				50	7		
12				55	6		
13		45.28889	=RISKGENERAL(0,60,{10,20,45,50,55},{1,6,8,7,6})				
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							
28							

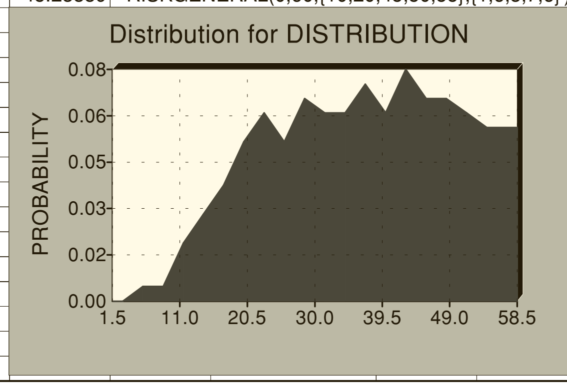


FIGURE 15

tween the given points, the density function changes at a linear rate. Thus, 30 would have a likelihood of

$$6 + \frac{(30 - 20)*(8 -6)}{(45 - 20)} = 6.8$$

Modeling Correlations

Suppose we have three normal random variables, each having mean 0 and standard deviation 1, correlated as follows:

- Variable 1 and variable 2 have .7 correlation.
- Variable 1 and variable 3 have a .8 correlation.
- Variable 2 and variable 3 have a .75 correlation.

To model this correlation structure, we use the =RISKCORRMAT command. Simply enter your correlation matrix somewhere in the worksheet. In Figure 16, we chose C27:E29.

	B	C	D	E	F	G	H
25							
26							
27		1	0.7	0.8			
28		0.7	1	0.75			
29		0.8	0.75	1			
30							
31	1	Variable 1	1.793028	risknormal(0,1,riskcorrmat(c27:e29,1))			
32	2	Variable 2	-0.449129	risknormal(0,1,riskcorrmat(c27:e29,2))			
33	3	Variable 3	-0.521328	risknormal(0,1,riskcorrmat(c27:e29,3))			

FIGURE 16

For each variable, type in front of the variable's actual distribution the syntax

=Actual Risk Function, RISKCORRMAT(Matrix, i)

Here, Matrix (C27:E29 in this case) indicates where the correlation matrix resides, and *i* is the column of the correlation matrix that contains the correlations for variable *i*. Thus, for variable 1, the correlations come from the first column of the correlation matrix.

If you run a simulation and extract the data for cells D31:D33, you will find that

- Each cell has a mean of around 0 and a standard deviation around 1.
- Each cell follows a normal distribution.
- D31 has around a .7 correlation with D32.
- D31 has around a .8 correlation with D33.
- D32 has around a .75 correlation with D33.

Truncating Random Variables

Suppose you believe that market share for a product is approximately normally distributed, with mean .6 and standard deviation .1. This random variable could exceed 1 or be negative, which would be inconsistent with the fact that market share must be between 0 and 1. To resolve this, you may enter the random variable from the Define Distribution icon as shown in Figure 17.

You could also type in formula

=RISKNORMAL(.6,.1,RISKTRUNCATE(0,1))

Then @Risk generates a normal random variable with mean .6 and standard deviation .1. If the random variable assumes a value between 0 (the lower truncation value) and 1 (the upper truncation value), that value is retained. Otherwise, another value is generated. The truncation values must be within 5 standard deviations of the mean.

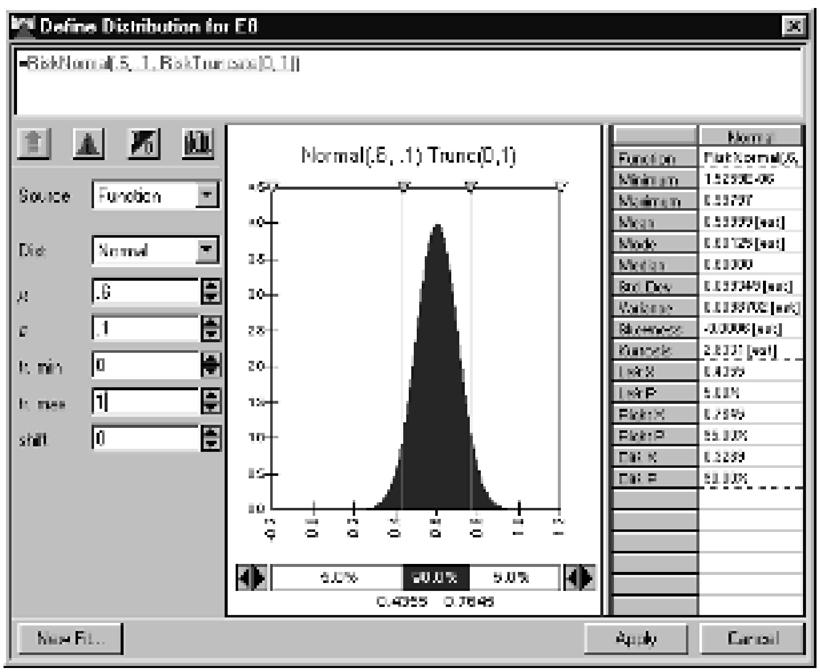


FIGURE 17

The RISKPERT Function

This function is similar to the RISKTRIANG function. The RISKPERT function is used to model the duration of projects. For example,

=RISKPERT(5,10,20)

would be used to model the duration of an activity that always takes at least 5 days, never takes more than 20 days, and is most likely to take 10 days. Whereas RISKTRIANG has a piecewise linear density function, the RISKPERT density has no linear segments. It is a special case of a Beta random variable.

Common Error Message

The error message "Invalid number of arguments" means that an incorrect syntax has been used with an @Risk function. For example, =RISKDUNIFORM({A1:A7}) may have been used instead of =RISKDUNIFORM(A1:A7).