In the name of Allah

Amirkabir University of Technology
(Tehran Polytechnic)

Industrial Engineering Department

Engineering Economics

Simulation using Excel

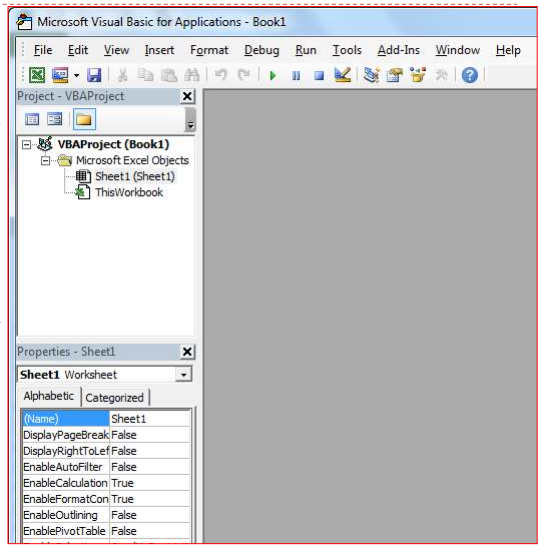
By: Akbar Esfahanipour

In this presentation

- How to write GetFormula Function
- Normal Distribution Functions in Excel
- Generating Random Numbers using Excel
- Computing π using Monte Carlo
- Some VBA notes

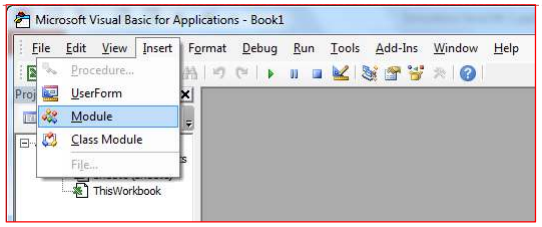
How to write GetFormula Function

- Getformula function
  - A small VBA program you can add to an Excel worksheet.
- Adding Getformula to your spreadsheet
  - Open the spreadsheet in which you want the formula to work.
  - Push [Alt] + F11. This will open the VBA editor something like:

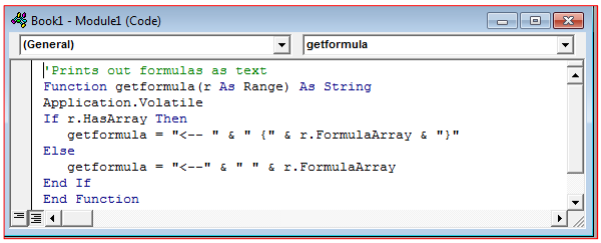


How to write GetFormula Function

- Hit **Insert|Module**



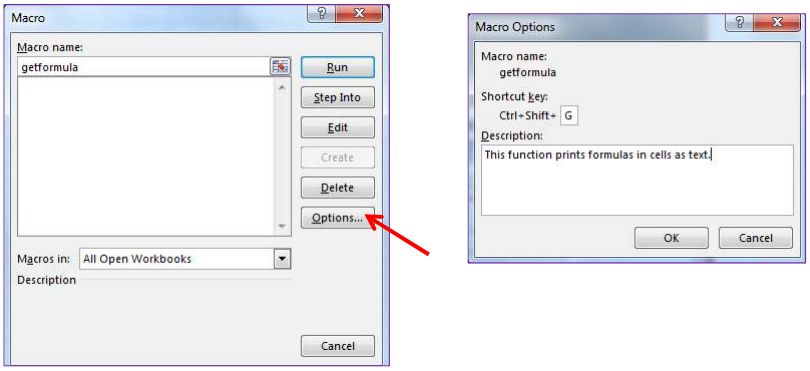
- Then insert the following text into the Module window



- Then **Save** the Module

How to write a help for your function

- How to write a help for your function:
  - In Excel 2010 or later:
    - View -> Macros -> View macros

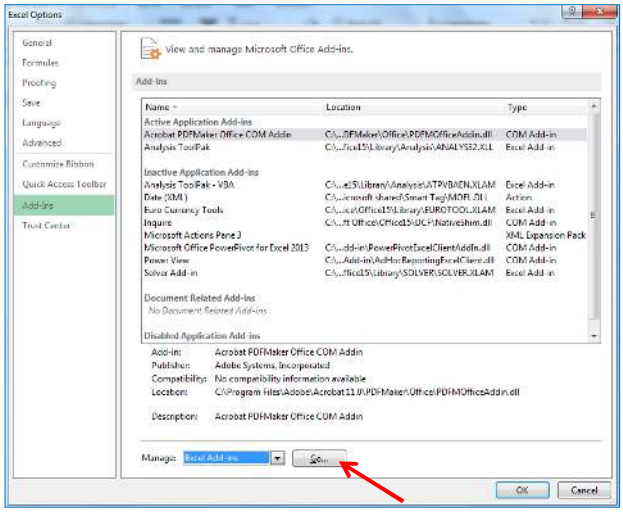


Normal Distribution Functions in Excel

- NORMDIST(x,mean, standard_dev, cumulative)
  - Returns the normal distribution for the specified mean and standard deviation.
- NORMINV(probability, mean, standard_dev)
  - Returns the inverse of the normal cumulative distribution for the specified mean and standard deviation.
- NORMSINV(probability)
  - Returns the inverse of the standard normal cumulative distribution.
- NORMSDIST(z)
  - Returns the standard normal cumulative distribution function.

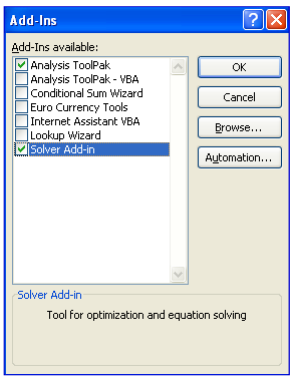
Generating Random Numbers using Excel

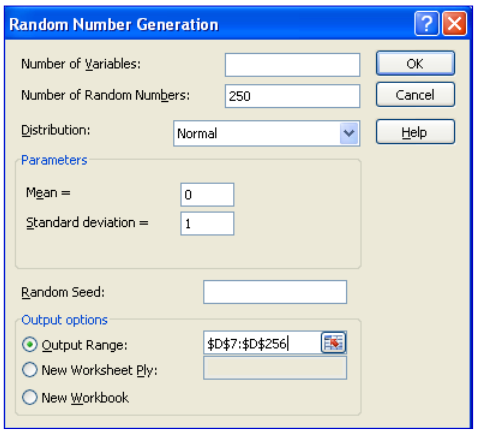
- Add Analysis ToolPak Add-ins
  - Go to Excel Options through File|Options|Add-Ins
  - Select the Analysis ToolPak Add-ins
  - Hit Go



Generating Random Numbers using Excel

- Add Analysis ToolPak from available Add-Ins
- Data | Data Analysis | Random Number Generation





Generating Random Numbers using Excel

- Functions for generating random numbers in Excel,
  - RAND() produces a random sample between 0 and 1.
  - NORMSINV(RAND()) produces a random sample from standard normal distribution.
  - NORMINV(RAND(), *m*, *v*) produces a random sample from a normal distribution with mean *m* and variance *v*.
  - RANDBETWEEN(*low*, *high*) function generates a random number between low and high

Generating Random Numbers using Excel

- To obtain two correlated standard normal samples
  - Obtain independent normal samples x_1 and x_2 from standard normal distribution and set

$$\varepsilon_1 = x_1$$
$$\varepsilon_2 = \rho x_1 + x_2\sqrt{1-\rho^2}$$

    - ρ: correlation coefficient between two data series
- For calculating correlation in Excel
  - Use correl() function
- For calculating Sample Variance and Sample Standard deviation in Excel
  - Use Var() function, use Varp() for population variance.
  - Use Stdev() function, use Stdevp() for population standard deviation.

Using VBA to Define some Functions

```
(General)                                   dOne
Function dOne(Stock, Exercise, Time, Interest, sigma)
    dOne = (Log(Stock / Exercise) + Interest * Time) / (sigma * Sqr(Time)) _
        + 0.5 * sigma * Sqr(Time)
End Function
Function CallOption(Stock, Exercise, Time, Interest, sigma)
    CallOption = Stock * Application.NormSDist(dOne(Stock, Exercise, _
        Time, Interest, sigma)) - Exercise * Exp(-Time * Interest) * _
     Application.NormSDist(dOne(Stock, Exercise, Time, Interest, sigma) _
      - sigma * Sqr(Time))
End Function
'Put pricing function uses put-call parity theorem
Function PutOption(Stock, Exercise, Time, Interest, sigma)
    PutOption = CallOption(Stock, Exercise, Time, Interest, sigma) _
         + Exercise * Exp(-Interest * Time) - Stock
End Function
```

Book2.xlsm - Module1 (Code)

```
(General)                                   summation
Function summation(range, start_range, end_range)
    Application.Volatile
    summation = 0
    For i = start_range To end_range
        summation = summation + range(i)
    Next i
End Function
```

Using VBA to Define some Functions

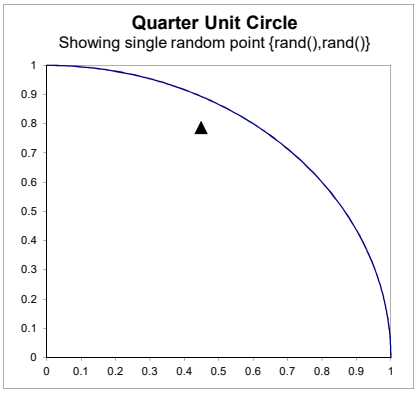
- Note the stop criteria
  - This always works because of the monotonic function.

```
(General)                                   CallVolatility
Function CallVolatility(Stock, Exercise, Time, Interest, Target)
    High = 1
    Low = 0
    Do While (High - Low) > 0.0001
    If CallOption(Stock, Exercise, Time, Interest, (High + Low) / 2) > _
        Target Then
            High = (High + Low) / 2
            Else: Low = (High + Low) / 2
    End If
    Loop
    CallVolatility = (High + Low) / 2
End Function
```

Computing π using Monte Carlo

- π : the area of a unit circle
  - π/4 : the area of a quarter circle
- To do this:
  - Inscribe a quarter circle into a unit square
  - Generate random points at the unit square
  - Count the relative number of points that fall inside the unit circle
  - Compute π: the area of the quarter circle * 4

Computing π using Monte Carlo



	A	B	C	D	E
1	**COMPUTING PI USING MONTE CARLO METHODS INITIAL EXPERIMENT**				
2	Number of data points	30	<-- =COUNT(A:A)		
3	Inside circle	22	<-- =COUNTIF(D:D,TRUE)		
4	Pi?	2.933333333	<-- =B3/B2*4		
5					
6		Each cell in these columns contains the Excel function =**Rand()**			
7	**Experiment**	**Random1**	**Random2**	**In unit circle?**	
8	1	0.43898	0.90591	FALSE	<-- =(B8^2+C8^2<=1)
9	2	0.73530	0.55413	TRUE	
10	3	0.83084	0.40433	TRUE	
11	4	0.87073	0.61628	FALSE	
12	5	0.31835	0.61798	TRUE	
13	6	0.13944	0.28215	TRUE	
14	7	0.87428	0.66875	FALSE	
15	8	0.96619	0.43453	FALSE	
16	9	0.75186	0.21914	TRUE	
17	10	0.59876	0.60792	TRUE	
18	11	0.42286	0.32925	TRUE	
19	12	0.32687	0.84465	TRUE	
20	13	0.30367	0.82461	TRUE	
21	14	0.23173	0.52777	TRUE	
22	15	0.96026	0.05040	TRUE	
23	16	0.84995	0.62523	FALSE	
24	17	0.05905	0.46049	TRUE	
25	18	0.51004	0.07220	TRUE	
26	19	0.90080	0.71615	FALSE	
27	20	0.78203	0.27752	TRUE	
28	21	0.61134	0.37597	TRUE	
29	22	0.70602	0.42085	TRUE	
30	23	0.43073	0.53358	TRUE	
31	24	0.89596	0.94230	FALSE	
32	25	0.13623	0.21620	TRUE	
33	26	0.80687	0.70309	FALSE	
34	27	0.36070	0.01285	TRUE	
35	28	0.16141	0.35535	TRUE	
36	29	0.89404	0.37844	TRUE	
37	30	0.07706	0.10867	TRUE	

Press F9 to give different results

Computing π using Monte Carlo

	A	B	C	D	E
1	**COMPUTING PI USING MONTE CARLO METHODS INITIAL EXPERIMENT**				
2	Number of data points	30	<-- =COUNT(A:A)		
3	Inside circle	24	<-- =COUNTIF(D:D,TRUE)		
4	Pi?	3.2	<-- =B3/B2*4		
5					

	A	B	C	D	E
1	**COMPUTING PI USING MONTE CARLO METHODS INITIAL EXPERIMENT**				
2	Number of data points	65,529	<-- =COUNT(A:A)		
3	Inside circle	51,676	<-- =COUNTIF(D:D,TRUE)		
4	Pi?	3.15438966	<-- =B3/B2*4		
5					

	A	B	C
1	**COMPUTING PI USING VBA**		
2	Number of data points	50,000,000	<-- This cell called "Number"
3	Pi?	3.14162752	<-- This cell called "Estimate"
4			
5			
6	StartTime	12:13:33	<-- This cell called "StartTime"
7	StopTime	12:14:07	<-- This cell called "StopTime"
8	Elapsed	0:00:34	<-- =Stoptime-StartTime

Computing π using Monte Carlo

```
Sub MonteCarlo()
    n = Worksheets("MC").Range("Number")
    Hits = 0
    For Index = 1 To n
        If Rnd ^ 2 + Rnd ^ 2 < 1 Then Hits = Hits + 1
  Next Index
  Range("Estimate") = 4 * Hits / n
End Sub
```

VBA notes

- User-Defined Functions with Visual Basic for Applications
  - see chapters 33-39 of the book Financial Modeling 3rd edition
- Range object and some properties
  - E.g., columns, rows
- Dim
  - For declaring the variables as a type (see next slide).
  - VBA arrays always start with index 0.
- Redim
  - Before you can use the array you need to set its size.
- Application.WorksheetFunction
  - For using the Excel functions
- For … Next statement
  - For repeating some commands

A Short List of VBA Types

Data Type	Range
Byte	0 to 255
Boolean	True or False
Integer	−32,768 to 32,767
Long (integer)	−2,147,483,648 to 2,147,483,647
Single (floating point)	−3.403E38 to −1.401E-45 for negative values; 1.401E-45 to 3.403E38 for positive values
Double (floating point)	−1.798E308 to −4.941E-324 for negative values; 4.941E-324 to 1.798E308 for positive values
Currency (scaled integer)	−922,337,203,685,477.5808 to 922,337,203,685,477.5807
Decimal	±79,228,162,514,264,337,593,543,950,335 with no decimal point; ±7.9228162514264337593543950 with 28 places to the right of the decimal
Date	January 1, 100 to December 31, 9999
String (variable length)	0 to approximately 2 billion
String (fixed length)	1 to approximately 65,400
Variant (with numbers)	Any numeric value up to the range of a Double
Variant (with characters)	Same range as for variable-length String

Define ranges in Excel

- Define ranges by
  - Select the range | Right click | Name a Range
- See ranges on
  - Formulas | Name Manager