Report on Monte Carlo Simulation

→ By Jai Katariya
→ Ms. in Business Analytics
→ University of Colorado, Denver

INDEX

1. Introduction

Monte Carlo simulation is a simulation that relies on repeated random sampling to perform statistical analysis to compute results and make inferences. This method of simulation is very closely related to random experiments, experiments for which the specific result is not known in advance. Thus, Monte Carlo simulation can be considered as what-if analysis.

What-if Analysis

We use mathematical models in natural sciences, social sciences, and engineering disciplines to describe the interactions in a system using mathematical expressions. Models depend on a number of input parameters, which are processed by the model's formulas, resulting in one or more outputs.

Figure 1: Mathematical models.

Image taken from www.informs-sim.org/wsc08papers/012.pdf

During the process, the input parameters depend on various external factors which are subjected to risk or variance. Here we make a deterministic model known as a base model, which doesn't involve these variations. Now, an effective model should be considerate of the risks associated with various input parameters. Thus, we generate various versions of the model, including the base model (with no risk/variation), the best possible scenario (with favorable variations) and the worst scenario (with the most unfavorable variations).

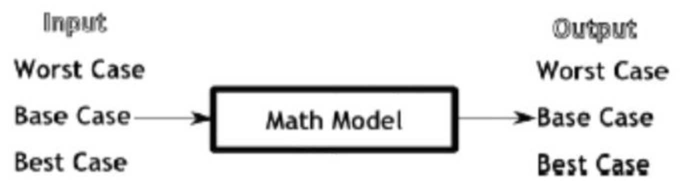


Figure 2: Case-based modeling.

Image taken from www.informs-sim.org/wsc08papers/012.pdf

Disadvantages of such an approach are:

1. It is possible we might not be able to evaluate the best and worst-case scenario.
2. All input variables may not be at their best or worst levels at the same time.
3. As we increase the number of cases, the model versioning and storing becomes difficult.
4. An experimenter might be tempted to run various ad-hoc values of the input parameters, often called what-if analysis, but it is not practical to go through all possible values of each input parameter.

Thus, Monte Carlo simulation can help to methodically investigate the complete range of risk associated with each input variable.

Basic process and steps in Monte Carlo simulation

1. We identify a statistical distribution for each of the input parameters.
2. We draw a random sample from each distribution, representing the value of each input parameter.
3. This is processed inside a system/model and returns an output/set of outputs.
4. The value of each output parameter is one particular outcome scenario in the simulation run.
5. Finally, to make a decision, we will perform statistical analysis on the values of the output parameters. For example, we can use the sampling statistics of the output parameters to characterize the output variation.

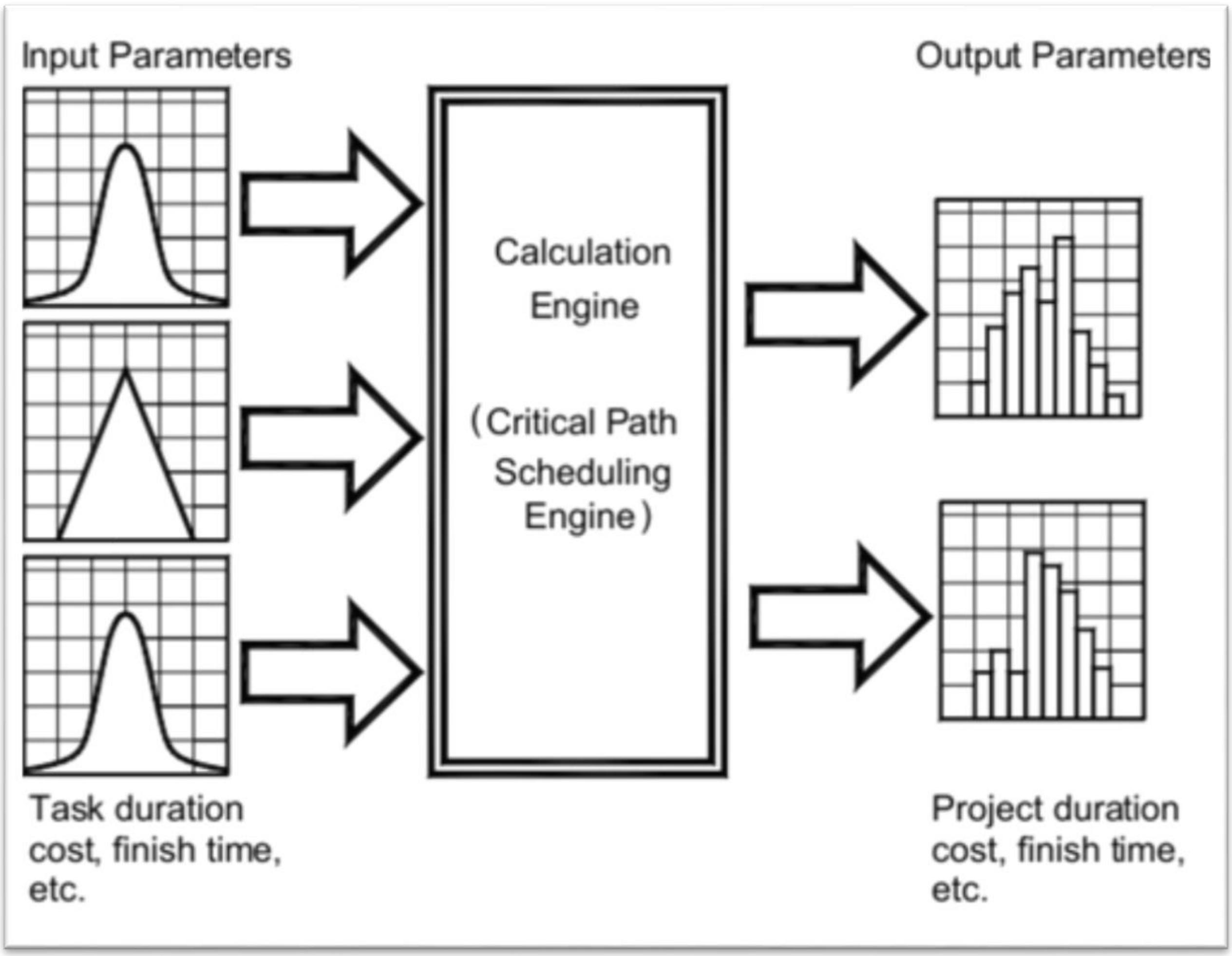


Image taken from https://www.slideshare.net/Intaver/monte-carlo-schedule-risk-analysis

2. Terminologies

1. Statistical distributions

Statistical distributions or probability distributions describe the outcomes of a varying random variable and the probability of occurrence of those outcomes. In simple terms, it is a description of a relative number of times each possible outcome will occur in a number of trials. The random variable can be discrete or continuous and depending on that, the probability distributions are discrete or continuous.

Few examples for **discrete** probability distributions are
1. Binomial distribution
2. Poisson distribution
3. Hypergeometric distribution.

Few examples for **continuous** probability distributions are
1. Normal distributions
2. Exponential distributions
3. Gamma distributions

2. Random sampling and Random number generator (RNG)

In statistics, when a finite subset of individuals from a population is drawn at random, it implies that each individual has an equal chance of being selected. This process of selecting the unit of population at random is done by a computational or physical device designed to generate a sequence of number that appears to be independent draws from the population. They are also called as Pseudo-random number generators since they are simulated. In this article, they used RNG to be uniformly distributed between 0 and 1.

3. Methodology

1. Static Model Generation

This is generally the very first step of every Monte Carlo simulation, developing a deterministic model with a close resemblance to the real scenario. As discussed before, we create the base model with the most likely values of the input parameters. This is the step of generating the static model, which returns a desired (most likely) output.

2. Input Distribution Identification

After identifying our deterministic model, we start adding risk components of each input parameters to the model. We identify the distributions followed by each input parameter via its historical data. This will be discussed more in section 4 of the paper.

3. Random Variable Generation (core of Monte Carlo simulation)

Once we get the distributions for each input variable, we draw random samples from these distributions. Each set of the random sample contains random values for each input variables, which will be added to our base model or deterministic model, providing a set of output. Now, the above process is iterated thousands of time to create a thousand sets of outputs. We will discuss this step in detail in section 5.

4. Analysis and Decision Making

Now we have a thousand sets of outputs from the simulations. Thus we perform a statistical analysis on them obtaining statistical confidence for us to make a decision. We will discuss this step briefly in section 6.

4. Identification of input distribution

In this section, we will discuss the procedure for identifying the input distributions for the simulation model, often called distribution fitting. We use numerical methods if we have the existing historical data for the input parameter (continuous or discrete). This provides a way to identify the most suitable probability distribution. Fitting routines are nonlinear optimization problems, variables are parameters of the distributions.

Methods for Distribution Fitting

1. Method of Maximum Likelihood (ML)

ML estimation (MLE) is a statistical method used to make inferences about parameters of the probability distribution from a given data set. The assumption made is that the data from a distribution is independent and identically distributed. It is the method of estimating the parameters by finding the parameter values that maximize the likelihood of making the observation. It is the most popular method to estimate the unknown parameters of a distribution. MLE bias tends to be negligible as the number of samples increases to infinity i.e. asymptotically unbiased. No unbiased estimator has lower mean squared error than thus, the MLE is asymptotically efficient.

2. Method of Moments (ME)

In this method, we estimate the population parameters such as mean, variance, etc. by equating sample moments with unobservable population moments and then solve those equations for the quantities to be estimated. Although ML estimates have a higher probability of being close to quantities to be estimated. But in some cases, the method of moments and the method of maximum likelihood are symbiotic. In some cases, generally with small samples, the method of moments are outside of the parameter space, thus making them unreliable.

3. Nonlinear Optimization

Another method for estimating the unknown parameters of a distribution is to use nonlinear optimization. The different objective functions that can be used are:
1. Minimizing one of the goodness-of-fit statistics,
2. Minimizing the sum-squared difference from sample moments (mean, variance, skewness, kurtosis), **or**
3. Minimizing the sum-squared difference from the sample percentiles (or quartiles or deciles).

To facilitate the optimization problem we can add additional constraints. But it is generally less efficient and takes more time.

Goodness-Of-Fit Statistics is the correctness of fitting a dataset to a distribution. Various software is used to decide the best fitting distribution. The three of the most common GOF statistics used are:

1. Chi-square Test

It tests how likely the observed distribution is due to chance. Assuming the variables are independent, it measures how well the observed distribution of data fits with the distribution.

The test statistic is given by the following equation.

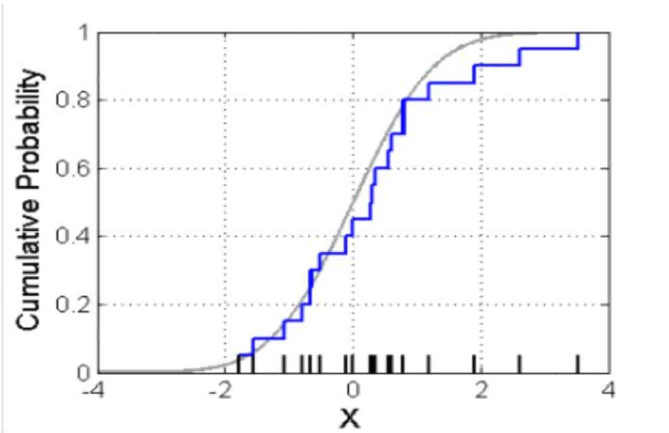
$$\hat{\chi}^2 = \sum_{j=1}^{k} \frac{(N_j - np_j)^2}{np_j}$$

Image taken from www.informs-sim.org/wsc08papers/012.pdf

2. EDF Statistic

In statistics, an empirical distribution function is the distribution function associated with the empirical measure of a sample. This cumulative distribution function is a step function that jumps up by 1/n at each of the ndata points. Its value at any specified value of the measured variable is the fraction of observations of the measured variable that are less than or equal to the specified value. The empirical distribution function estimates the cumulative distribution function underlying of the points in the sample and converges with probability 1.

Above Definition and the below Image taken from: https://en.wikipedia.org/wiki/Empirical_distribution_function



The blue line shows an empirical distribution function. The black bars represent the samples corresponding to the empirical distribution function and the gray curve is the true cumulative distribution function.

An example of Distribution Fitting

The paper takes a dataset of the weight of 30 students and tries to fit normal and lognormal distributions. For the normal distribution, the mean and standard deviation is calculated.

	Mean/Log-mean	**Standard deviation/Log-SD**	**Anderson-Darling (AD) statistic**	**p-value**
Normal Distribution	149.16	25.97	0.1763	0.919
Lognormal Distribution	4.99	0.18	0.1871	0.828

Normal distribution was better among the two fits due to two reasons:

1. The Anderson-Darling (AD) statistic that was calculated was lower for the normal as compared to lognormal distribution.
2. Moreover, when we compared the p-values of the two fitted distributions, normal distribution was a better fit since it had larger p-value indicating better fit.

5. Random variable generation

Once we have identified the underlying distributions for the input parameters of a simulation model, we generate random numbers from these distributions (discrete or continuous). The most direct route for generating a random sample from a distribution is provided by the inverse transformation method. After inversing the probability density function (PDF) (for continuous distributions) or probability mass function (PMF) (for discrete distributions), it is converted to a random value between 0 and 1.

Advantages of inverse transformation method:

1) It can be used to generate random numbers from a truncated distribution.
2) Negative correlation can be successfully induced between two variables, since this method preserves the monotonicity between the uniform variate U and the random variable X.
3) This method can be used for discrete, continuous and functions which are a mixture of discrete and continuous distributions.

Disadvantages of inverse transformation method:

1) Difficult to implement if there is no closed-form inverse CDF for a distribution.

Bootstrapped Monte Carlo

When there is no/few historical data for the input parameters, which doesn’t allow us to obtain an underlying distribution for an input, we can use Bootstrapped Monte Carlo (MC) simulation also known as bootstrapping, to generate random variables. In statistics, bootstrapping is a process where random sampling is done with replacement to estimate statistical parameters. It allows assigning the measure of accuracy to our sample estimates. In simple terms, it is resampling the same sample over and over which provides accurate results.

Disadvantages of Bootstrapping

1. The tendency to be overly optimistic.
2. May violate the assumption of independence of samples.
3. Fails to account the correlation in repeated observations which leads to false statistical significance.

6. Monte Carlo simulation output analysis

The result of the Monte Carlo simulation depends on the statistical analysis performed on the output. The various statistical analysis are:

1. Computing the mean of the trial output values, thus achieving the expected value of the output.
2. Perform Exploratory Data Analysis like creating frequency histogram which provides the approximate shape of the probability density function of the output variable.
3. These output values can be used as empirical distribution to calculate percentiles and other statistics.
4. Lastly, they can be fitted to a probability distribution to develop confidence bands.

6.1 Formulas for Basic Statistical Analysis

Mean $(\bar{x})$

$$\bar{x} = \frac{1}{n}\sum_i x_i$$

Median 50th percentile

Standard Deviation (s)

$$s = \sqrt{\frac{1}{N-1}\sum_i (x_i - \bar{x})^2}$$

Variance (s^2)

$$s^2 = \frac{1}{N-1}\sum_i (x_i - \bar{x})^2$$

Skewness

$$\text{Skewness} = \frac{\sum_i (x_i - \bar{x})^3}{(N-1)s^3}$$

Kurtosis

$$\text{Kurtosis} = \frac{\sum_i (x_i - \bar{x})^4}{(N-1)s^4} - 3$$

Coeff. of Variability

$$\text{Coeff. of Variability} = \frac{s}{\bar{x}}$$

Minimum (x_{min})

$$x_{min} = \min_i \; x_i$$

Maximum (x_{max})

$$x_{max} = \max_i \; x_i$$

Range Width

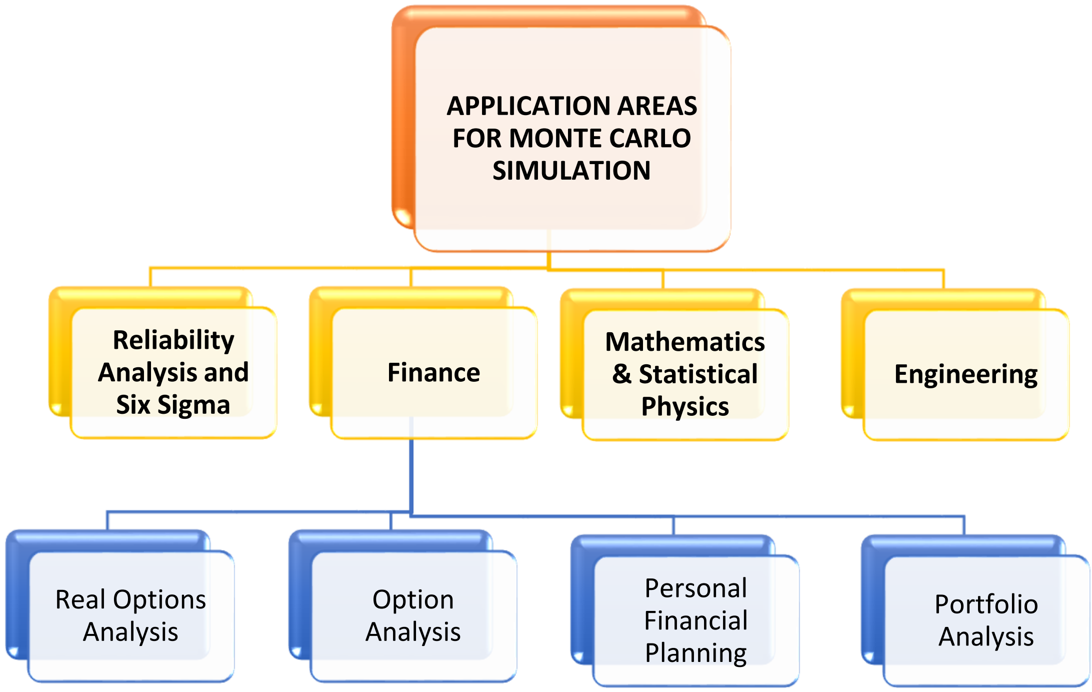
$$\text{Range Width} = x_{max} - x_{min}$$

Mean Std. Error

$$\text{Mean Std. Error} = \frac{s}{\sqrt{n}}$$

Image taken from www.informs-sim.org/wsc08papers/012.pdf

7. <u>Application areas for Monte Carlo simulation</u>



7.1 Monte Carlo Simulation in Finance

Monte Carlo simulation is used by many analysts, in finance, to model scenarios. For example:

1. **Real Option Analysis** - stochastic models use MC simulation to characterize a project's net present value (NPV). This model will then incorporate the variability associated with the input into the model and run Monte Carlo simulations to calculate the average NPV of a potential investment.

2. **Portfolio Analysis / Risk Analysis** – Portfolios are evaluated by Monte Carlo methods to generate 1000s of iteration which can be displayed in a histogram to get the portfolio probability distribution. Later on, we can perform statistical analysis like mean, median, percentile, etc.

3. **Option Analysis** - MC simulation can analyze the financial instruments like options by generating various alternative price paths for the options. Similarly, bond options with uncertain annualized interest rates can be simulated by MC analysis.

4. **Personal Financial Planning** – Simple Monte Carlo simulation is done on the market to figure out the probability of attaining the target balance for retirement.

7.2 Monte Carlo Simulation in Reliability Analysis and Six Sigma

Here, we are dealing with the capabilities of a system to perform under a given set of conditions and evaluating the failure patterns. Six sigma is a business management strategy, which tries to identify and remove causes of failures and errors. MC simulation can help the six-sigma efforts by identifying optimal strategy, estimating probabilities for project cost benefit, creating virtual testing grounds, predicting quality of business processes, identifying defect-producing process steps driving unwanted variation etc.

7.3 Monte Carlo Simulation in Mathematics and Statistical Physics

From complex multi-dimensional partial differentiation to optimization in operations or quantum systems in quantum domains, simulation is key method used to solve them

7.4 Monte Carlo Simulation in Engineering

Most common uses of MC simulation are:

1. Mechanical Engineering to estimate reliability of mechanical components
2. Chemical engineering to analyze effective life of pressure vessels in reactors.
3. Electronics engineering and circuit design where the chips are simulated
4. Computer science and software engineering

8. Monte Carlo simulation software

Various options are available to use basic Monte Carlo simulations in computers. There are many programs which are possibly tailor-made for specific situations. Various software libraries are also available in most of these high level programming languages, to run MC simulations. MC simulations can be performed in a spreadsheet software like Microsoft Excel as well as high-level programming language like C, C++, Java, R. After using and running the simulation we are capable of generating charts and graphs of the output to gain some insights for further analysis.

To demonstrate, I will run a simple Monte Carlo simulation on investments till retirement in two software:

1. Microsoft Excel
2. R

9A. <u>Monte Carlo Simulation in Microsoft Excel</u>

Steps:

a. Inputs

To demonstrate the Monte Carlo simulation of an investment portfolio, I have assumed the following:

1) Investment - $100,000 in S&P500
2) Average of 11.2% a year
3) Standard deviation of volatility of 18%.
4) Years till retirement – 30
5) Additional investment $10,000 per year.

→ My aim is to know what my investment will be worth in 30 years at my retirement.

Monte Carlo Simulation On Investment

Current Investment	$100,000.00		**Mean**		
Average Return	11.20%		**Median**		
St. Deviation of return	18%		**St Deviation**		
Time to retirement	30		**Percentilies**		
Amount to invest Annually	$10,000.00		**5%**		
			25%		
Ending Value (single Iteration)					
			Year	**Return**	**Ending Value (add investment at end of the year)**
Simulation Iteration			1		
1			2		
2			3		

b. Generating Random Returns by 'NORM.INV' function

→ I have generated a random rate of return for one year from today, by using the norm inverse function. Here, I have assumed that the rate of return follows a normal distribution with an average return of 11.2% and has a standard deviation of 18%.

SUM | fx =NORM.INV(RAND(),B3,B4)

	A	B	C	D	E	F	G
1			Monte Carlo Simulation On Investment				
2	Current Investment	$100,000.00		Mean			
3	Average Return	11.20%		Median			
4	St. Deviation of return	18%		St Deviation			
5	Time to retirement	30		Percentilies			
6	Amount to invest Annually	$10,000.00		5%			
7				25%			
8	Ending Value (single Iteration)						
9							
10				Year	Return	Ending Value (add investment at end of the year)	
11	Simulation Iteration			1	=NORM.IN		

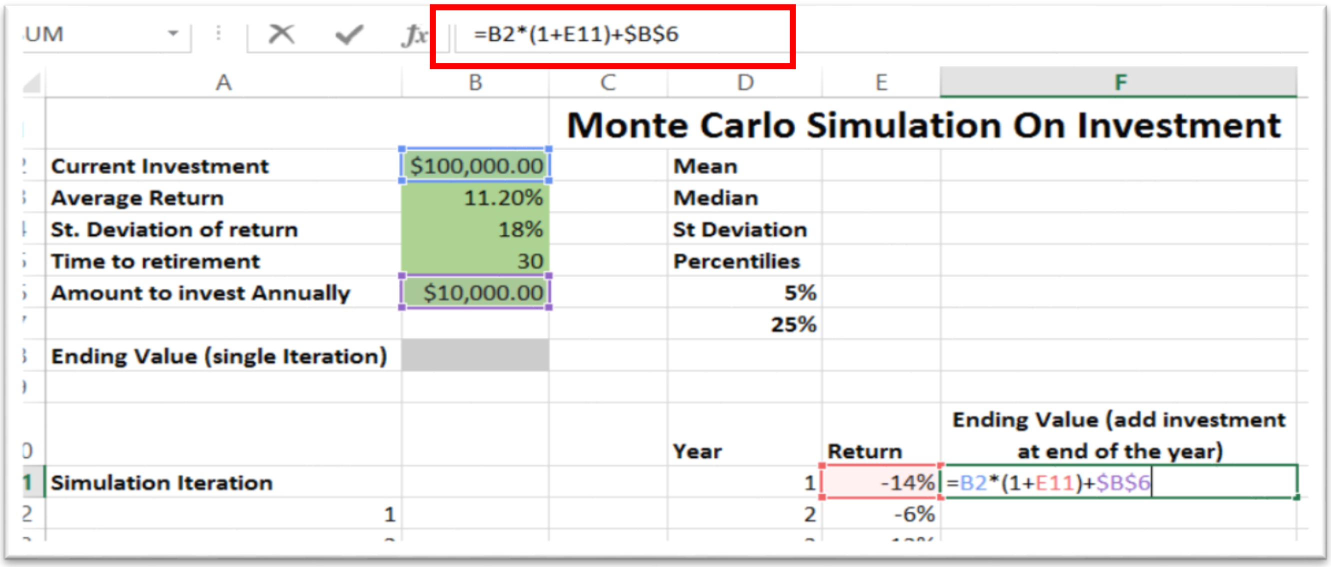
→ Using the norm inverse function, we can see that one possible return is over -14%. By generating another return, I received -6%.

		Monte Carlo Simulation On Investment			
Current Investment	$100,000.00		Mean		
Average Return	11.20%		Median		
St. Deviation of return	18%		St Deviation		
Time to retirement	30		Percentilies		
Amount to invest Annually	$10,000.00		5%		
			25%		
Ending Value (single Iteration)					
			Year	Return	Ending Value (add investment at end of the year)
Simulation Iteration			1	-14%	
1			2	-6%	
2			3	12%	
3			4	-2%	

c. Calculating the Ending Values for every year

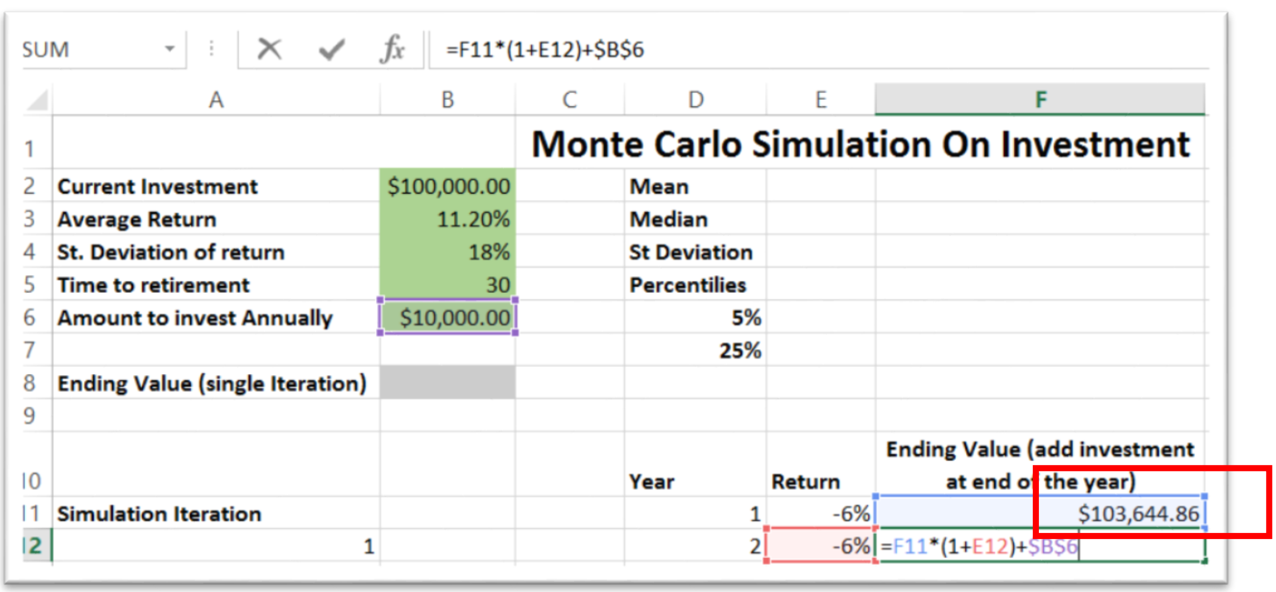
→ 1st year ending value calculated

Having attained a return, we move to the ending balance. Now, there are a number of ways to generate the ending balance. In my example, I have taken the beginning balance, multiplied it by 1 and added the annual rate of return. This product is then added to annual investment ($10000) which is assumed to be at the end of year. This is how I generated one possible ending value.



→ Other ending value calculated

In the second year we will replace the initial investment with the first years ending value. To calculate for 30 years, I simply copied down the formula from the second year to get a stream of possible returns in 30 years.



→ So one possible outcome we have is ending with about 3.35 million dollars.

				investment at end of the year)
		Year	Return	
imulation Iteration		1	24%	$134,204.01
1		2	19%	$170,190.66
2		3	12%	$200,306.75
3		4	18%	$247,011.94
4		5	10%	$280,983.03
5		6	-13%	$253,898.46
6		7	-22%	$207,989.42
7		8	27%	$274,692.22
8		9	21%	$341,363.60
9		10	-5%	$334,235.80
10		11	14%	$391,594.36
11		12	-6%	$379,381.23
12		13	0%	$390,060.20
13		14	0%	$399,342.69
14		15	52%	$615,006.41
15		16	52%	$942,443.91
16		17	20%	$1,139,110.84
17		18	23%	$1,412,980.40
18		19	-3%	$1,377,497.79
19		20	21%	$1,670,290.16
20		21	-7%	$1,558,151.44
21		22	-15%	$1,334,805.47
22		23	26%	$1,690,438.72
23		24	-12%	$1,491,429.57
24		25	-7%	$1,400,679.27
25		26	24%	$1,750,435.62
26		27	19%	$2,097,714.35
27		28	15%	$2,420,036.49
28		29	16%	$2,811,640.32
29		30	19%	$3,352,446.27

→ Another one is ending with about 5 million

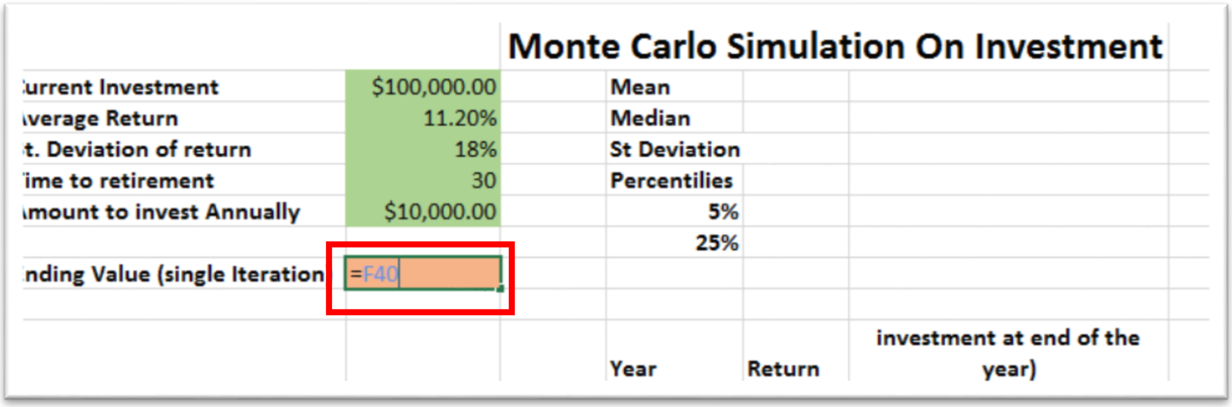
28	24%	$4,351,897.02
29	10%	$4,811,738.52
30	4%	$5,032,050.29

→Then there's 1.15 million and so on. Thus, proving the random function.

28	19%	$1,082,862.93
29	17%	$1,279,382.33
30	-12%	$1,131,699.56

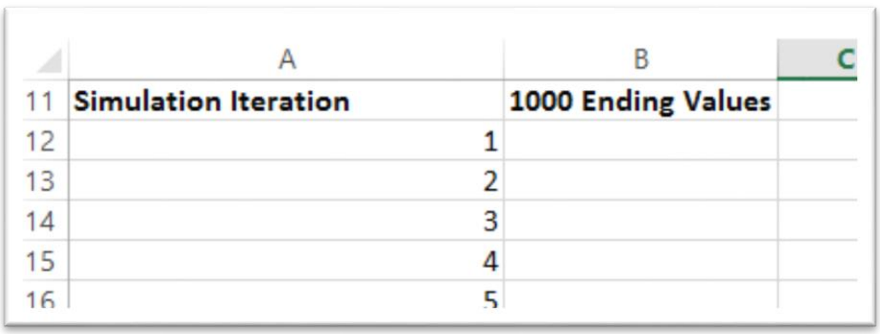
d. Simulating the investment value 1000 times

→ As we don't know the future, we are not sure of which one to select. Thus, we are going to simulate the ending value at 30th year and generate 1000 possible ending values. This is Monte Carlo simulation. Firstly, let's reference the one ending value that we got from f40 cell.

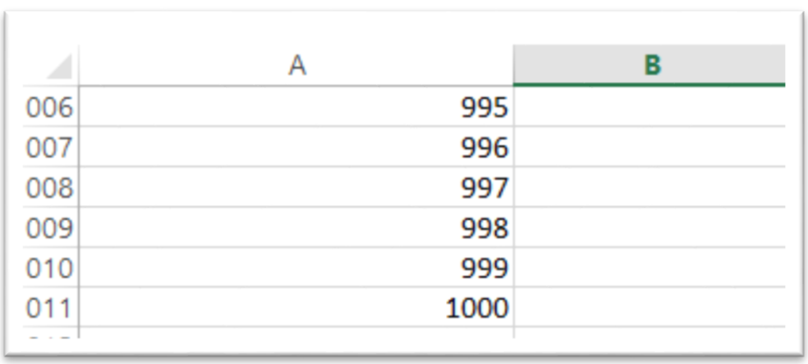


		Monte Carlo Simulation On Investment		
urrent Investment	$100,000.00	Mean		
verage Return	11.20%	Median		
t. Deviation of return	18%	St Deviation		
ime to retirement	30	Percentilies		
mount to invest Annually	$10,000.00	5%		
		25%		
nding Value (single Iteration	=F40			
		Year	Return	investment at end of the year)

→ I've set up a table where we will get the thousand iterations to get a reasonable understanding of my investment to be at my retirement.

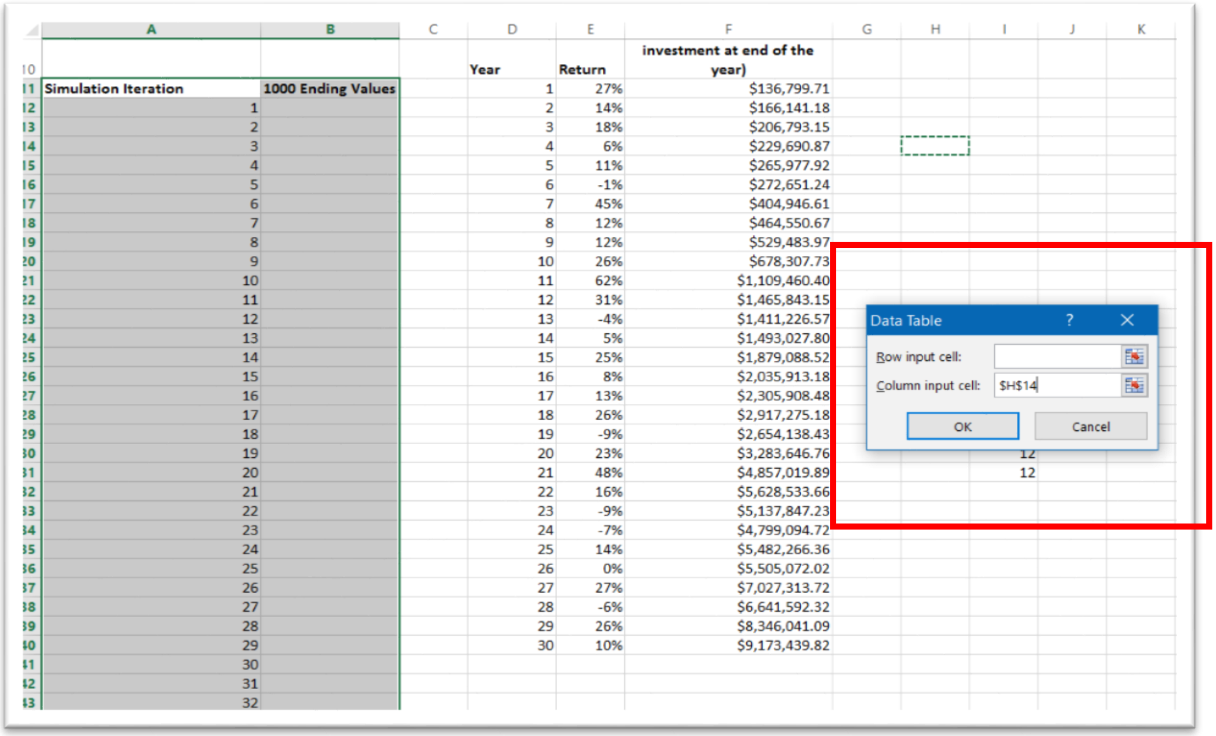


	A	B	C
11	Simulation Iteration	1000 Ending Values	
12	1		
13	2		
14	3		
15	4		
16	5		

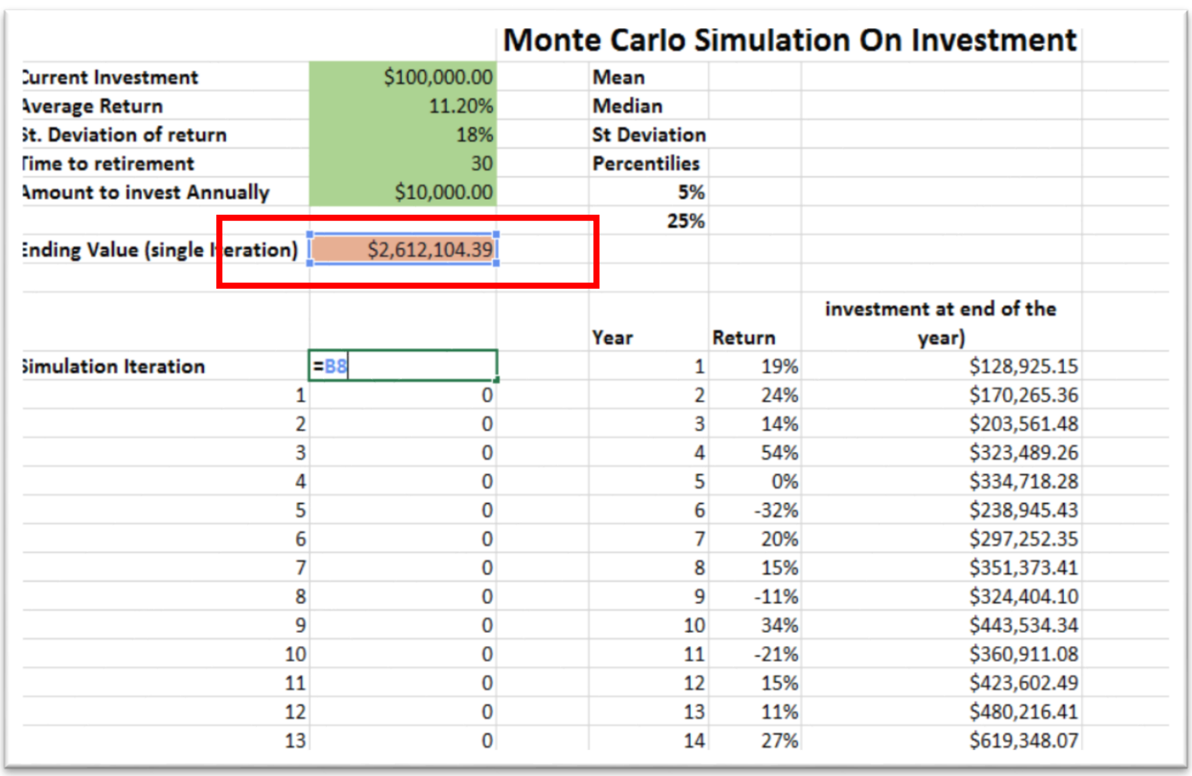


	A	B
006	995	
007	996	
008	997	
009	998	
010	999	
011	1000	

→ By selecting the whole range, we use the **Data Tool and Data Table** but it is not the conventional way, instead I just take a blank cell so there is no input.



Simulation Iteration	1000 Ending Values	Year	Return	investment at end of the year)
1		1	27%	$136,799.71
2		2	14%	$166,141.18
3		3	18%	$206,793.15
4		4	6%	$229,690.87
5		5	11%	$265,977.92
6		6	-1%	$272,651.24
7		7	45%	$404,946.61
8		8	12%	$464,550.67
9		9	12%	$529,483.97
10		10	26%	$678,307.73
11		11	62%	$1,109,460.40
12		12	31%	$1,465,843.15
13		13	-4%	$1,411,226.57
14		14	5%	$1,493,027.80
15		15	25%	$1,879,088.52
16		16	8%	$2,035,913.18
17		17	13%	$2,305,908.48
18		18	26%	$2,917,275.18
19		19	-9%	$2,654,138.43
20		20	23%	$3,283,646.76
21		21	48%	$4,857,019.89
22		22	16%	$5,628,533.66
23		23	-9%	$5,137,847.23
24		24	-7%	$4,799,094.72
25		25	14%	$5,482,266.36
26		26	0%	$5,505,072.02
27		27	27%	$7,027,313.72
28		28	-6%	$6,641,592.32
29		29	26%	$8,346,041.09
30		30	10%	$9,173,439.82
31				
32				



Monte Carlo Simulation On Investment

Current Investment	$100,000.00	Mean	
Average Return	11.20%	Median	
St. Deviation of return	18%	St Deviation	
Time to retirement	30	Percentiles	
Amount to invest Annually	$10,000.00	5%	
		25%	
Ending Value (single Iteration)	$2,612,104.39		

Simulation Iteration	=B8	Year	Return	investment at end of the year)
1	0	1	19%	$128,925.15
2	0	2	24%	$170,265.36
3	0	3	14%	$203,561.48
4	0	4	54%	$323,489.26
5	0	5	0%	$334,718.28
6	0	6	-32%	$238,945.43
7	0	7	20%	$297,252.35
8	0	8	15%	$351,373.41
9	0	9	-11%	$324,404.10
10	0	10	34%	$443,534.34
11	0	11	-21%	$360,911.08
12	0	12	15%	$423,602.49
13	0	13	11%	$480,216.41
		14	27%	$619,348.07

→ And then what it will do in turn is just generate a thousand different ending values.

			Monte Carlo Simulation On Investment		
Current Investment		$100,000.00	Mean		
Average Return		11.20%	Median		
St. Deviation of return		18%	St Deviation		
Time to retirement		30	Percentiles		
Amount to invest Annually		$10,000.00	5%		
			25%		
Ending Value (single Iteration)		$4,156,565.08			
			Year	Return	investment at end of the year)
Simulation Iteration		$4,156,565.08	1	22%	$131,714.37
	1	10747981.4	2	2%	$144,933.88
	2	4600940.874	3	5%	$161,595.95
	3	3356002.482	4	37%	$230,801.98
	4	917608.5353	5	11%	$265,792.67
	5	10706273.17	6	17%	$321,586.53
	6	2271947.07	7	5%	$348,470.56
	7	2870307.992	8	33%	$472,713.35
	8	3207866.277	9	19%	$570,714.43
	9	2452947.191	10	14%	$662,273.72
	10	3215005.708	11	37%	$915,601.78
	11	5002748.78	12	49%	$1,378,532.14

→ **Currency Formatting**

	A		B	C	D
1				Monte Carl	
2	Current Investment		$ 100,000.00		Mean
3	Average Return		11.20%		Median
4	St. Deviation of return		18%		St Deviat
5	Time to retirement		30		Percentil
6	Amount to invest Annually		$ 10,000.00		
7					2
8	Ending Value (single Iteration)		$ 4,156,565.08		
9					
10					Year
11	Simulation Iteration		**$ 4,156,565.08**		
12		1	$ 10,747,981.40		
13		2	$ 4,600,940.87		
14		3	$ 3,356,002.48		
15		4	$ 917,608.54		
16		5	$ 10,706,273.17		
17		6	$ 2,271,947.07		

→ Thus, we get our thousand possible ending values for our portfolio.

1004	993.00	$	9,631,652.11
1005	994.00	$	4,947,771.34
1006	995.00	$	1,473,736.92
1007	996.00	$	1,319,247.49
1008	997.00	$	5,030,776.93
1009	998.00	$	1,757,113.59
1010	999.00	$	7,447,974.68
1011	1,000.00	$	4,683,943.01

e. Performing Statistical Analysis

1. Average – it is not an average portfolio but it is central tendency number.

=AVERAGE(B12:B1012)

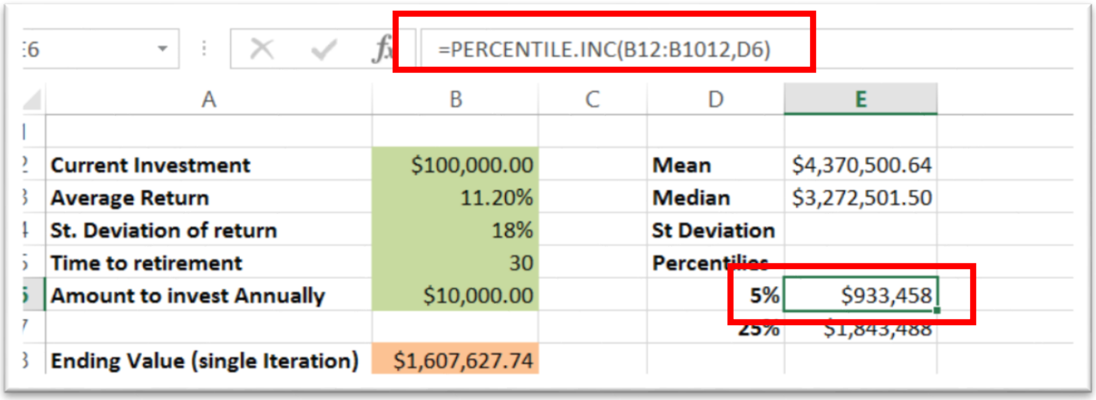
A	B	C	D	E	F
Current Investment	$100,000.00		Mean	$4,636,096.05	
Average Return	11.20%		Median		
St. Deviation of return	18%		St Deviation		
Time to retirement	30		Percentiles		
Amount to invest Annually	$10,000.00		5%		
			25%		
Ending Value (single Iteration)	$1,859,046.42				

2. Median – to check if this distribution is skewed or no. The mean is much higher than the median we can see that this distribution is **very positively skewed**.

=MEDIAN(B12:B1012)

A	B	C	D	E	F
Current Investment	$100,000.00		Mean	$4,520,064.91	
Average Return	11.20%		Median	$3,420,805.92	
St. Deviation of return	18%		St Deviation		
Time to retirement	30		Percentiles		
Amount to invest Annually	$10,000.00		5%		
			25%		
Ending Value (single Iteration)	$1,762,153.38				

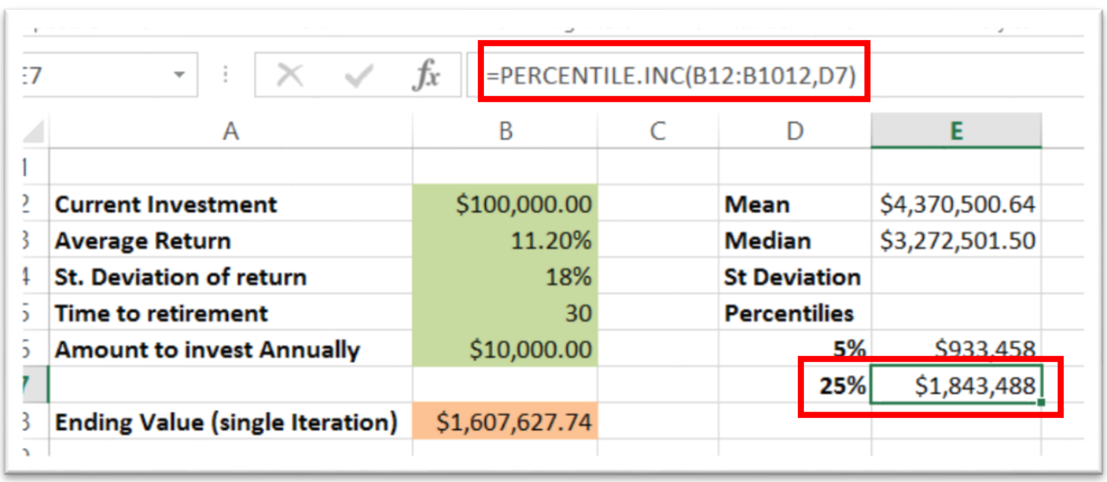
3. Percentile - Make probability statements by calculating percentiles. We use percentile inclusive function on fifth percentile to get 964000. This means that there is a ninety-five percent chance that we will have somewhere more than nine hundred sixty four thousand dollars at the end of 30 years if returns continue as they sort of historically have at 11.2 and eighteen percent standard deviation.



=PERCENTILE.INC(B12:B1012,D6)

	A	B	C	D	E
2	Current Investment	$100,000.00		Mean	$4,370,500.64
3	Average Return	11.20%		Median	$3,272,501.50
4	St. Deviation of return	18%		St Deviation	
5	Time to retirement	30		Percentiles	
6	Amount to invest Annually	$10,000.00		5%	$933,458
7				25%	$1,843,488
8	Ending Value (single Iteration)	$1,607,627.74			

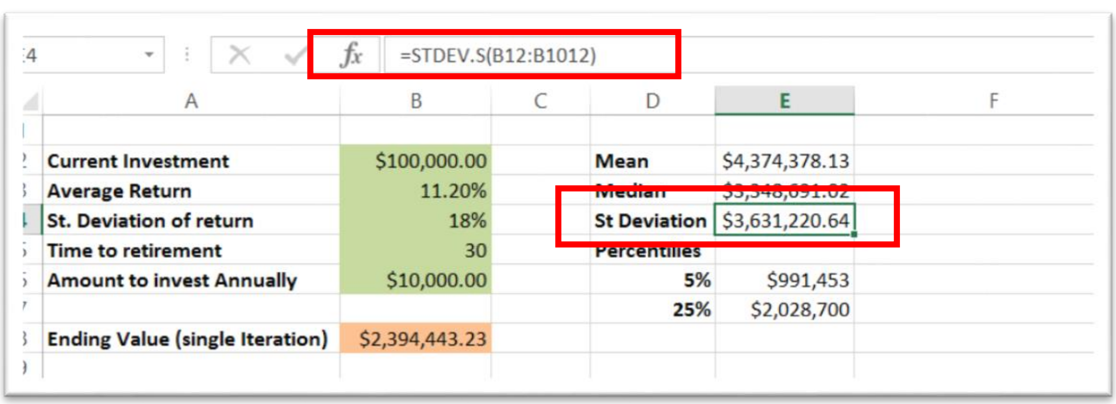
→ We also found 25th percentile. We can say there is a twenty-five percent chance of having somewhere less than 1.7 million or seventy-five percent chance of having something more than 1.7 million.



=PERCENTILE.INC(B12:B1012,D7)

	A	B	C	D	E
2	Current Investment	$100,000.00		Mean	$4,370,500.64
3	Average Return	11.20%		Median	$3,272,501.50
4	St. Deviation of return	18%		St Deviation	
5	Time to retirement	30		Percentiles	
6	Amount to invest Annually	$10,000.00		5%	$933,458
7				25%	$1,843,488
8	Ending Value (single Iteration)	$1,607,627.74			

4. Standard Deviation – to check how volatile or how spread out our distribution is, we can use standard deviation function.



=STDEV.S(B12:B1012)

	A	B	C	D	E	F
2	Current Investment	$100,000.00		Mean	$4,374,378.13	
3	Average Return	11.20%		Median	[illegible]	
4	St. Deviation of return	18%		St Deviation	$3,631,220.64	
5	Time to retirement	30		Percentiles		
6	Amount to invest Annually	$10,000.00		5%	$991,453	
7				25%	$2,028,700	
8	Ending Value (single Iteration)	$2,394,443.23				

9B. Monte Carlo Simulation in R

→ We use 'RNORM' function to generate random annual returns for time to retire years. Here we calculate the return on investment for time to retire for each year and we compound the returns after each year till the time of retirement. Now we get a list of returns up to time of retirement.

The formula for this is mentioned below**:**

```
current_Investment  = 100000
Average_Return = 0.112
Standard_deviation = 0.18
Time_to_retire = 30
Annual_investment =10000

Retirement_value = NULL

for (k in 0:1000){
random_returns_for_next_30_years = rnorm(n = Time_to_retire, mean = Average_Return, sd =
Standard_deviation)
random_returns_for_next_30_years

Ending_Value = NULL
first_ending_value = 0
for (i in 1:length(random_returns_for_next_30_years)){
  if(i == 1){
    first_ending_value = current_Investment*(1+random_returns_for_next_30_years[i])+Annual_investment
  }
  if(i != 1){
    first_ending_value = (first_ending_value*(1+random_returns_for_next_30_years[i]))+Annual_investment
    Ending_Value[i] = first_ending_value
  }
}
Retirement_value[k] = round(Ending_Value[Time_to_retire])
}
```

This above process is now simulated 1000 times to get a list of 1000 values that contains 1000 ending values at retirement at different random rate of returns. This helps to incorporate all possible values of returns in our analysis, which makes Monte Carlo simulation so powerful and yet so easy to grasp.

→ Performing Statistical Analysis

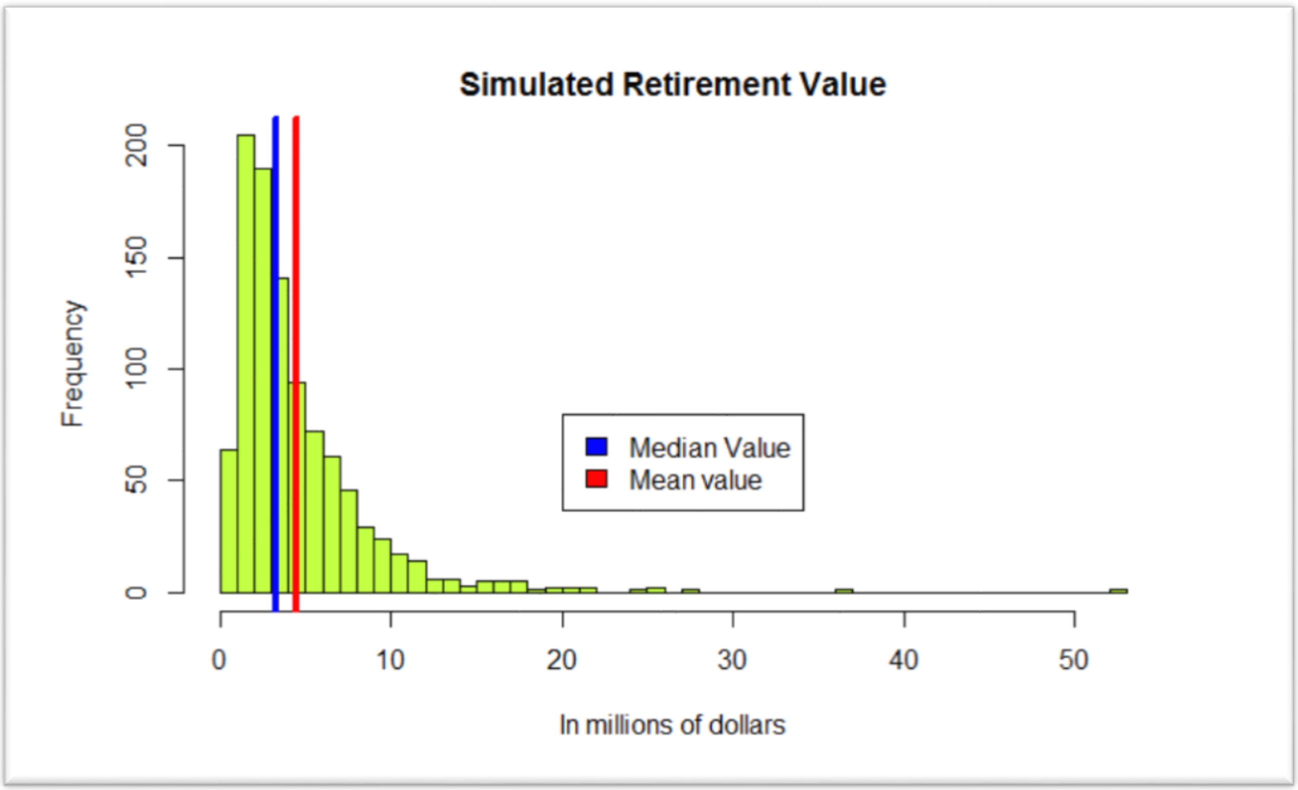
A histogram is drawn to show the above results with the mean and median marked to show that the distribution is positively skewed.

The formula for this is mentioned below:

```
median_value = median(Retirement_value)
median_value/1000000
mean_value = mean(Retirement_value)
mean_value/1000000

hist(Retirement_value, xlab = 'In millions of dollars',col = 'olivedrab1', breaks = 50, main = 'Simulated
Retirement Value')
abline(v = median_value, col = 'blue', lwd=4)
legend(x=20,y=80,legend=c('Median Value', 'Mean value'),fill=c("Blue", 'Red'))
abline(v = mean_value, col = 'red', lwd=4)
```

Histogram with mean and median marked.



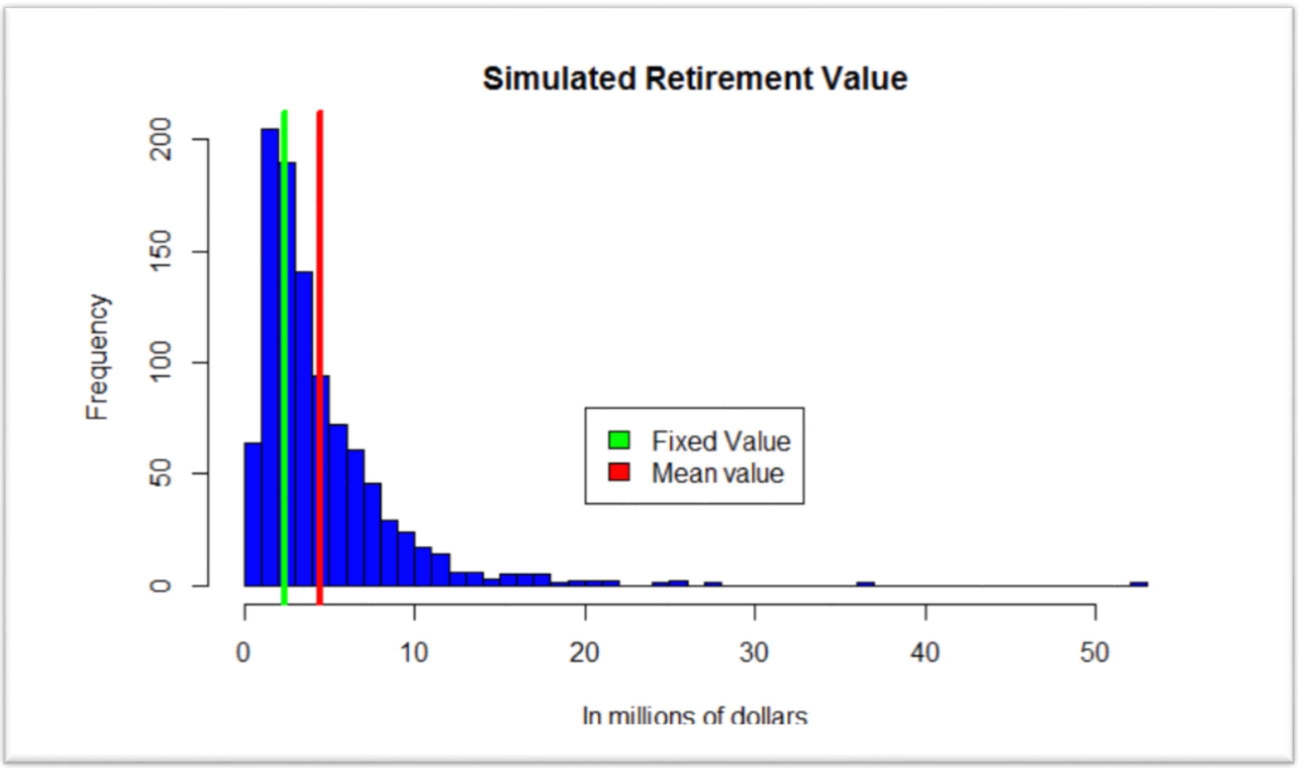
9C. Suggestions and Further Analysis

We can try to compare this simulated results to a particular fixed return for 30 years. This would help us know whether to invest in the fixed return or invest in an asset with an average return of 11.2% and standard deviation of 18%. To help us understand better we can take the average return as a fixed rate so that our comparison is at par. But we notice after the calculations that even making the average return fix is way too low than the mean of thousand iteration. Our investment will be 2.4 million with a fixed rate for 30 years and mean of our simulation suggest around 4.3 million.

The formula for this is mentioned below**:**

```
Fixed.rate =(100000*(1+.112)**30)/1000000
Fixed.rate
hist(Retirement_value, xlab = 'In millions of dollars',col = 'blue', breaks = 50, main = 'Simulated
Retirement Value')
abline(v = Fixed.rate, col = 'green', lwd=4)
legend(x=20,y=80,legend=c('Fixed Value', 'Mean value'),fill=c("Green", 'Red'))
abline(v = mean_value, col = 'red', lwd=4)
```

Histogram with mean and fixed rate value marked.



10. Conclusion

Monte Carlo simulation not only solves a statistical problem by a stochastic approach but also the simulation helps to graphical represent the problem. When combined together it allows us to understand the distribution of results for any statistical problem with inputs sampled over thousands of time. It is even possible to create an independent relationships between input variables.

When creating a financial plan for a client, the major issue is uncertainty. Monte Carlo simulation is a very powerful mathematical technique to analyze and generate probabilities to solve those uncertainty to certain extend. This allows analysts to convert investment chances into choices. It is simple and very easy in practice. To enhance our results and solve more complex problems various software are developing the methods of Monte Carlo simulation.

One of disadvantage is that Monte Carlo simulation has the ability to factor all possible values of inputs but it’s not always the case. The assumptions need to be fair because the output is only as good as the inputs. Another disadvantage is that it underestimates the extreme values, events. It also doesn’t factor in the behavioral aspects, irrationalities exhibited by people.

It is universally applicable, as we have noticed in session 7, when combined with different tools, the results are way more convincing than ever. Monte Carlo simulation combined with risk analysis is a powerful tool. It not only generates default probabilities but also produces expected creditor losses. Thus, this technique was used to quantity the default risk as well as the ability of creditors to recover the losses. It even works brilliantly with real options analysis, helps us to understand how finance distributes value and risk among project stakeholder.

11. References

a. Papers

→ inside.mines.edu/~gdavis/Papers/IJFERM.pdf

→ www.informs-sim.org/wsc08papers/012.pdf

b. Videos - For Monte Carlo simulation in Microsoft Excel

→ https://www.youtube.com/watch?v=UeGncSFijUM

→ https://www.youtube.com/watch?v=Q5Fw2IRMjPQ

c. Links

→ http://www.investopedia.com/articles/investing/112514/monte-carlo-simulation-basics.asp

→ https://www.advisorperspectives.com/articles/2014/08/26/the-power-and-limitations-of-monte-carlo-simulations

→ https://www.youtube.com/watch?v=Q5Fw2IRMjPQ

→ https://www.slideshare.net/Intaver/monte-carlo-schedule-risk-analysis

→ https://en.wikipedia.org/wiki/Empirical_distribution_function

~THANK YOU~