

Calculating Uncertainty

PROBABILITY MANAGEMENT WITH SIP MATH

5 January 2013

Marc Thibault

marc@smpro.ca

In *The Black Swan*, Nassim Nicholas Taleb has a marvelously compact definition of a probability distribution:

Nassim Nicholas
Taleb, *The Black
Swan* (2010), p.36.

" ... the model used to calculate the odds of different events, how they are distributed."

A SIP is like that – a model that connects events to their probabilities.

In statistical speak, a probability distribution is a platonic concept, an ideal – a class, not an instance. Probability distributions have names like ‘Gaussian’, ‘Weibull’, and ‘Beta’, not ‘MonthlyDemand’. They have well-defined formulas and parameters to plug into those formulas. A lot of calculus is involved.

Inconvenient probability distributions – instances of what Jacob Wolfowitz refers to as “the non-parametric case” – are assumed to have formulas and parameters back there somewhere, but no one can figure out what they are. Even more calculus is involved.

J. Wolfowitz,
*Annals of
Mathematical
Statistics XIII*
(1942), p.264.

In a remarkably simple way, a SIP can make the connections between events and their probabilities. No calculus is involved.

SIPs don't have names like 'Gaussian' or 'Cauchy' – they have names like 'MonthlyDemand' and 'TimeToComplete'.

In this monograph, I'll explain what a SIP is, then how SIPs and SIP math are used to calculate uncertainty and, along the way, how we go about composing SIPs with which to calculate.

SO WHAT'S A SIP?

When Taleb refers to the odds of different events, the 'events' are things like 'this project will finish in less than ten months'. That event has some probability of actually happening.

On the other hand, finishing within twelve months also has a probability, as does finishing within eight months. The project has many possible finish times, each with its own probability of being met. The project's finish date is an *uncertain variable*. It's a variable because it can have different values, and it's uncertain because its value is not predetermined.

One important concept: In most cases, the real-world uncertain variables we'll be working with don't have One True Value. More information, finer instruments, smarter people, none of these can eliminate the uncertainty, because the uncertainty is inherent. The roll of a die doesn't have one true value – it has six possible values. The next roll has one value, the one after that probably another. The time to complete a project doesn't have one true value, nor does the price of Apple shares.

Inherent
Uncertainty

Sam Savage proposed Stochastic Information Packets – SIPs – representing uncertain variables as part of the cure for The Flaw of Averages. He coined “The Flaw of Averages” to express what happens when you ignore the uncertainty in your data and calculate with averages or some other single number.

Sam L. Savage,
The Flaw of
Averages (2009),
p.332.

The result of Sam’s work is *Probability Management*, a new discipline with pragmatic tools and techniques for calculating uncertainty – with SIPs at the core.

A SIP is a data structure that holds a list of values. The values are from the possible values of an uncertain variable. The SIP for the roll of a die has the numbers 1 through 6. A SIP has the same relationship to an uncertain variable as a plain number does to an ordinary variable.



Suppose you were to record how long it takes to get to work, every morning for a thousand workdays. Your time to work changes every day and it’s relatively hard to predict; it’s an uncertain variable and a SIP is the best way to characterize the uncertainty. In Excel, or another application that supports vectors (aka arrays, lists), those thousand time values can be packaged into a SIP with a thousand elements.

A SIP in time

If you printed it out, it might look like this:

20,37,30,62,46,108,72,17,27,50,73,45,37,33,49,33,
46,32,31,18,29,32,53,96,39,90,21,96,33,25,92,39,
74,72,36,64,21, ...

extending out to a thousand comma-separated values.

At this point, someone with statistical training might try to fit the values to a famous-name probability distribution. We won’t do that. A SIP isn’t an incomplete thing crying out to be fitted into the platonic form of some distribution.



If it must have a named distribution we can say it has a *Sample Distribution*. I use that name because the values in the SIP will come from sampling the subject's possible values. The thousand travel times are a sample taken from all the possible travel times. We can also refer to each value as a sample, so your TimeToWork SIP is a 1000-sample SIP.

The Sample Distribution

Any subset of the possible values of an uncertain variable, including just one, qualifies as a sample.

By design, a SIP falls squarely into Wolfowitz's 'non-parametric case'. It's not that it can't be approximated with a well-defined famous-name probability distribution; it might be possible, but we don't care. In the world of SIPs, all distributions are sample distributions and all are equally simple.

We'll ***let the data speak for itself*** without the eye-crossing calculus. We're going to do that in part because we aren't looking for the One True Value; we're looking for insight into the range of values you might get and how likely various values are. Letting the data speak for itself means that all the values we work with to get that insight are valid, because they've actually happened at least once.

Let's say that you want to know the probability of getting to work in forty minutes or less. All you have to do is count the number of samples whose values are forty minutes or less. Now divide that count by 1000, the total number of samples. Lets say the count is 450. The SIP says you got to work in forty minutes or less 45% (450/1000) of the time. If nothing relevant has changed and history is a reasonably good guide to the future, there's a 45% probability, a little less than even odds, that you can make it to work in forty minutes or less tomorrow – and there's a 55% probability that you won't.

What are the odds?

The distinction between 'sample' and 'value' is important. The SIP has 1000 samples, but it can have less than 1000 different values. In this case it does – a

Samples vs Values

lot of the travel times repeat. Values close to more likely times repeat more often. Ideally, each of the n samples in a sample distribution has a probability of about $1/n$ of being closest to the next observed value (tomorrow's trip to work).

That's how a SIP is composed to make the connection between the values and the probabilities of an uncertain variable. The probability that the variable falls into a particular range is proportional to the number of samples whose values fall into that range. In this example, the range is from the minimum travel time to 40 minutes, and there were 450 samples in that range.

The condition "or less" makes the 45% a *percentile*. The percentile associated with a value is the percentage of samples whose values are equal to or less than that value. The 100th percentile is the SIP maximum. All the values are less than or equal to the maximum. The SIP minimum always has a percentile larger than zero – how much depends on the percentage of samples sharing the minimum value. In the case of the minimum, equal to or less is the same as just equal to. And then, the 0th percentile is undefined because every sample is associated with a non-zero percentile.

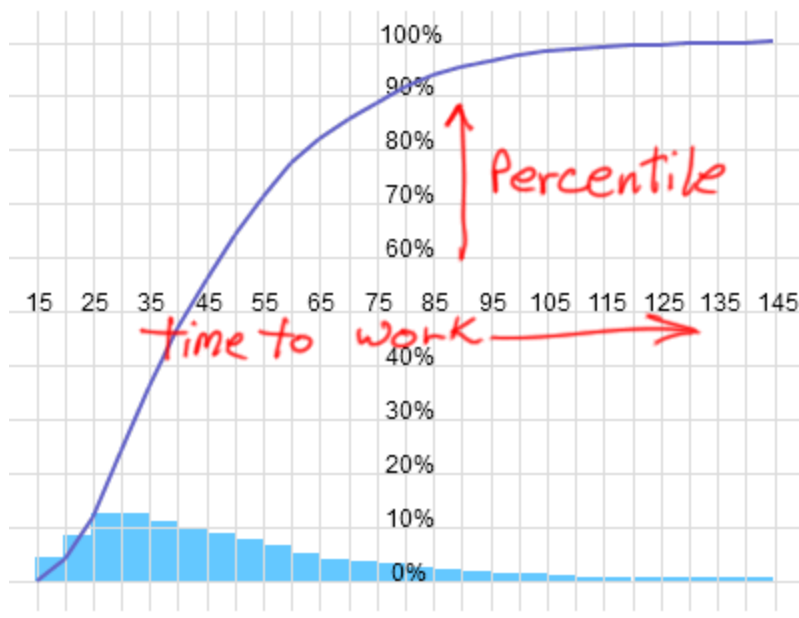
The 50th percentile is also known as the median. It relates to even odds, with as many samples above it as below.

There's a problem I glossed over: Asking specific questions about values and probabilities will get tedious. We need a way to get a richer view of the uncertain variable.

What we do is plot a graph that tells us a lot about the SIP. The characteristic graph for the TimeToWork SIP might look like this:

Percentiles express "or less" and "or more".

Median and average have a useful relationship. If the median value is higher than the average, a majority of samples will be above average. If the median is lower, a majority of samples will be below average.



A Percentile Chart

The vertical axis is percentile and the horizontal axis is time to work in minutes.

The first thing it tells us is that your travel times, so far, range from 15 to 145 minutes.

The dark blue line is the percentile curve. That's the "or less" graph. You can see where it crosses 40 minutes that the percentile is about 45%. That's a 45% probability of getting to work in 40 minutes or less.

Now let's turn the question around. Say you had a very important meeting tomorrow and you wanted to be sure to get to the office on time for the meeting. How much time should you allow to have a 90% probability, 9:1 odds, in favor of getting there on time?

The answer's on the percentile curve where it crosses the 90% line. You'd want to give yourself between 75 and 80 minutes. That doesn't guarantee that you'll make it on time – nothing can do that. It just reduces the risk of being late to 10%.

Percentile as the Probability of Success

Is making the meeting on time important enough to justify leaving for work 40 minutes earlier than usual?

When you take the inconvenience into account, you might settle on a different percentile. The percentile curve won't make a decision for you, but it will help you make an informed decision.

Another way of looking at a SIP is to look at its histogram – the light blue rectangular bars at the bottom of the chart. Each bar represents a range of values. The height of each bar is proportional to the number of samples whose values fall within its range. In this case, you can see the most likely travel times, those that happen most often, are between 25 and 35 minutes and have the highest bars. The less likely times have shorter bars.

We tend to think of the histogram as the SIP's *shape* and this shape tells us a lot about the variable. In this case, we can see that the most likely travel time is around 30 minutes and that it ranges from a little less to a lot more. You can get that from the percentile curve, but it's harder to see. That's why I usually plot both.

A histogram is the uncertain variable's *shape*.

Notice that the right hand side of the histogram trails off into the distance. Like many (most?) real-world uncertainties, the outcome can be much more extreme than what we've experienced to date. There are lots of low-probability events that could take you well past 145 minutes to work, including a crash scene that freezes traffic for a few hours or having a heart attack and not getting to work at all. How you deal with this 'long tail' is a context-sensitive issue beyond the scope of this discussion, but Taleb has some good thinking on the subject.

Beware the long tail.

If I asked how long (in this example) it takes to get to work, you might respond with "around 40 minutes", that being the time you generally allow. If I'm asking you for a lift, I might not be happy with the odds in favor of getting to work on time. Forty minutes is close

to the median – the “heads we’re on time, tails we’re late” point.

This is why “we can do this project in ten months” is a misleading oversimplification. When other people are depending on our projections, they deserve to know more.

SO WHAT’S SIP MATH?

SIP math is calculating with uncertain variables the way ordinary math calculates with single values. That is, it’s calculating uncertainty with SIPs.

SIP Math Defined

There are three reasons for working with probabilities. The first is to study probability itself as a subject. The second is to make general, largely theoretical, statements about classes of real or imagined events. The third is to make empirical statements about specific real-world events. The third is where SIP math is useful and we make no claims about the other two.

Let’s start with a simple Flaw of Averages situation:

You’re a grocer and one of your products is ready-to-serve barbecued chicken. Health regulations require that you discard any birds you don’t sell the day they were cooked. They cost you \$6 each and you sell them for \$9.

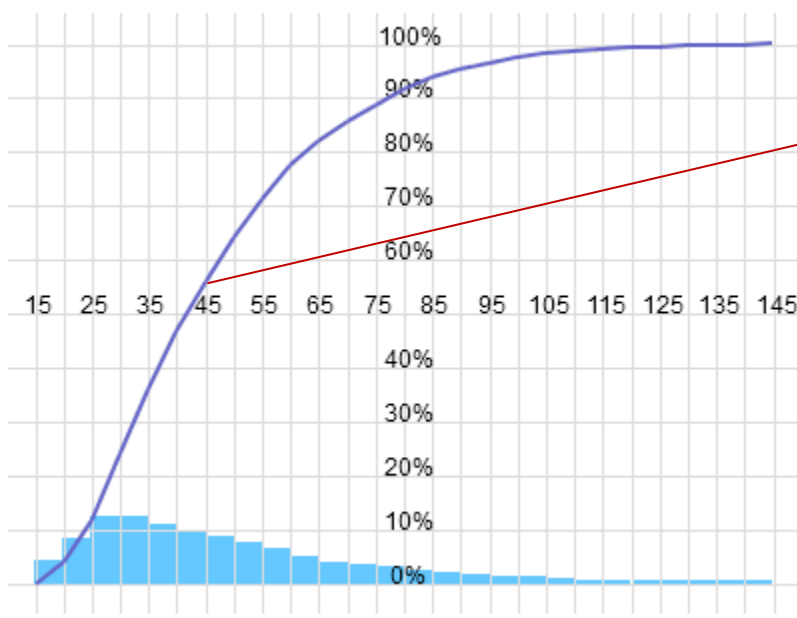
Let’s say we gathered data about the demand for cooked chickens by polling a random selection of customers as they came in the door over a period of a few months, and the key statistics are

Minimum	15
Maximum	142
Average	46
Median	41

From the SIP, you can calculate the average daily demand. In this case it's 46 chickens. Note that the median is lower than the average. This means that more than half the time the demand is below average.

Logic that says, "on average the demand will be 46 chickens, and you'll prepare 46 chickens, so on average you'll sell 46 chickens, and on average you'll see a \$138 gross profit," is classic, if somewhat obvious, Flaw of Averages fallacy. An average isn't enough. You need the SIP.

$$46 \times (9 - 6) = 138$$



Demand for 45 chickens or less 55% of the time

If you prepare 46 chickens, you'll have some left over more than half the time. The rest of the time you won't have enough to meet demand. If you prepare 46 birds, 46 isn't the average daily sales, it's the maximum. The average will be less.

On average, the average is wrong.

Knowing that, how many barbecued chickens is the daily optimum – producing the greatest expected profit?

To answer this question, we need a business model, something like this:

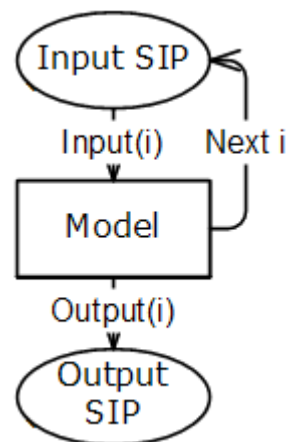
UnitCost	\$6.00	Minimum of stock and demand
UnitPrice	\$9.00	
Demand	46	
Stock	46	Stock × Unit Cost
Sales	46	Sales × Unit Price
Excess	0	
StockCost	\$276.00	
SalesRevenue	\$414.00	
GrossProfit	\$138.00	

The *Stock* is the number of chickens prepared and *Sales* is the minimum of *Stock* and *Demand*; you can't sell more chickens than you prepared, or more than the demand. But *Demand* is a SIP, not just a number, and its uncertainty is passed on to *Sales*. So *Sales* must be a SIP also. Since *Sales* is a SIP, *SalesRevenue* must be a SIP and *GrossProfit* must be a SIP. Not to worry, a bit of SIP math takes care of it all.

SIP in, SIP out.

Take a guess at the number of prepared chickens that will give you the highest expected gross profit.

What you need is the *GrossProfit* SIP. To get it, you get a computer to simulate 1,000 business days with a different *Demand* sample for each day. It'll go through the *Demand* SIP one sample at a time and use that sample's value as input to the business model. The model calculates gross profit. For each *Demand* sample value in, you get one sample value for the *GrossProfit* SIP coming out the bottom, until you've accumulated 1000 gross profit samples.



The fundamental SIP math formula is

$$\text{For all } i, \text{ outputs}(i) = \text{model}(\text{inputs}(i))$$

Inputs and outputs are plural because you can have more than one of each.

You may recognize this as being a bit like Monte Carlo Simulation but without the random number generators – without the ‘Monte Carlo’ – just the simulation.

In virtual space, you simulate 1,000 days of chicken sales – a thousand *trials* of your business model – and record the gross profit for each trial.

In general, for trial *i*,

$$\begin{aligned} \text{GrossProfit}(i) &= \text{SalesRevenue}(i) - \text{StockCost} \\ \text{SalesRevenue}(i) &= \text{Sales}(i) * \text{UnitPrice} \\ \text{Sales}(i) &= \text{minimum}(\text{Demand}(i), \text{Stock}) \\ \text{StockCost} &= \text{Stock} * \text{UnitCost} \end{aligned}$$

A trial is one numbered pass through the model.

That's the model.

Of course, you'll automate this in a way that lets you try different values for *Stock* and see what the effect is.

Using Excel and keeping it simple, you might have something that looks like this:

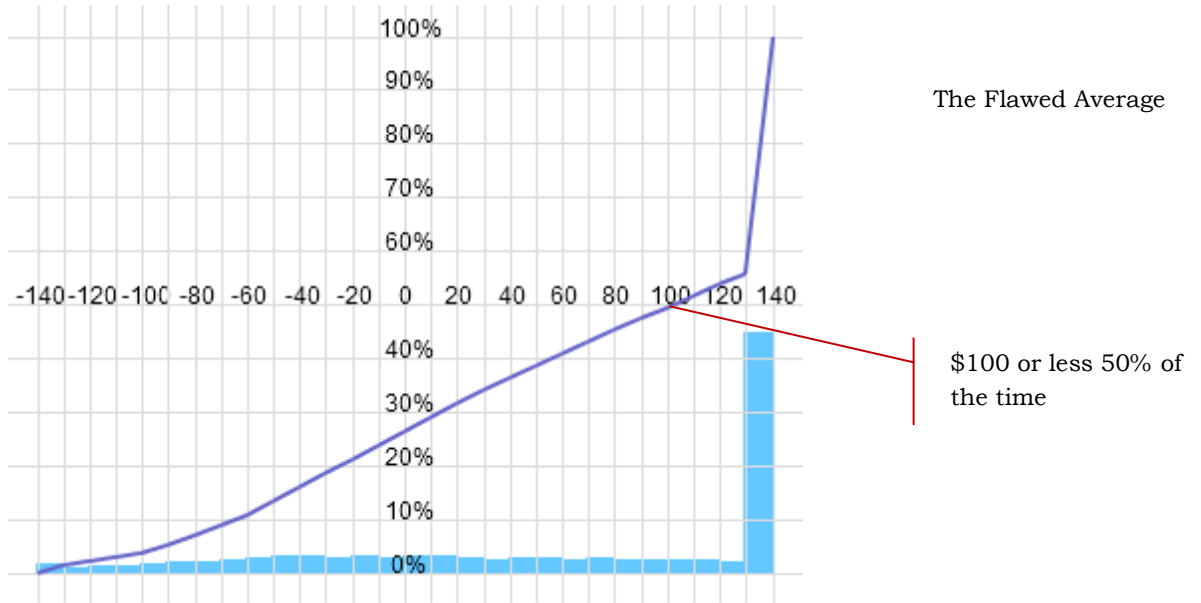
	<i>Single Trial #8</i>	<i>Expected Values</i>	Trial Index	Stock Level
UnitCost	\$6.00		▲	▲
UnitPrice	\$9.00		▼	▼
Demand	17	46	8	46
Stock	46			
Sales	17			
Excess	29			
StockCost	\$276.00			
SalesRevenue	\$153.00	\$339.38		
GrossProfit	-\$123.00	\$63.38		

Get the Excel workbook CalcUA01.xlsx at smpro.ca/sipmath.

The Trial Index spinner is there mainly to satisfy your curiosity. You can use it to step through some trials to see what effect each *Demand* value has on the other variables in the *Trial* column. The *Stock Level* spinner lets you easily change the number of birds you roast every day and see the effect on the expected gross profit. Notice that the expected value is a result only. You don't calculate with it.

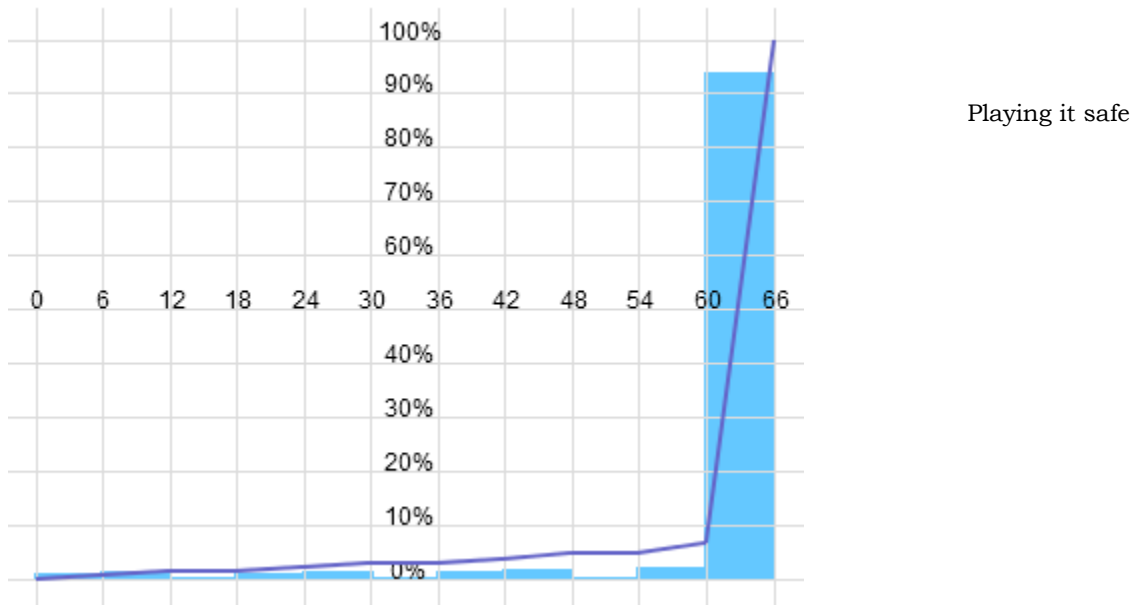
Stepping through the *Demand* values and recording the *GrossProfit* values is all happening in the background of whatever software tool set you use.

To see how this all works out, here's the *GrossProfit* SIP with Stock set at 46 chickens:



The expected gross profit is \$63.38 and you lose money 25% of the time.

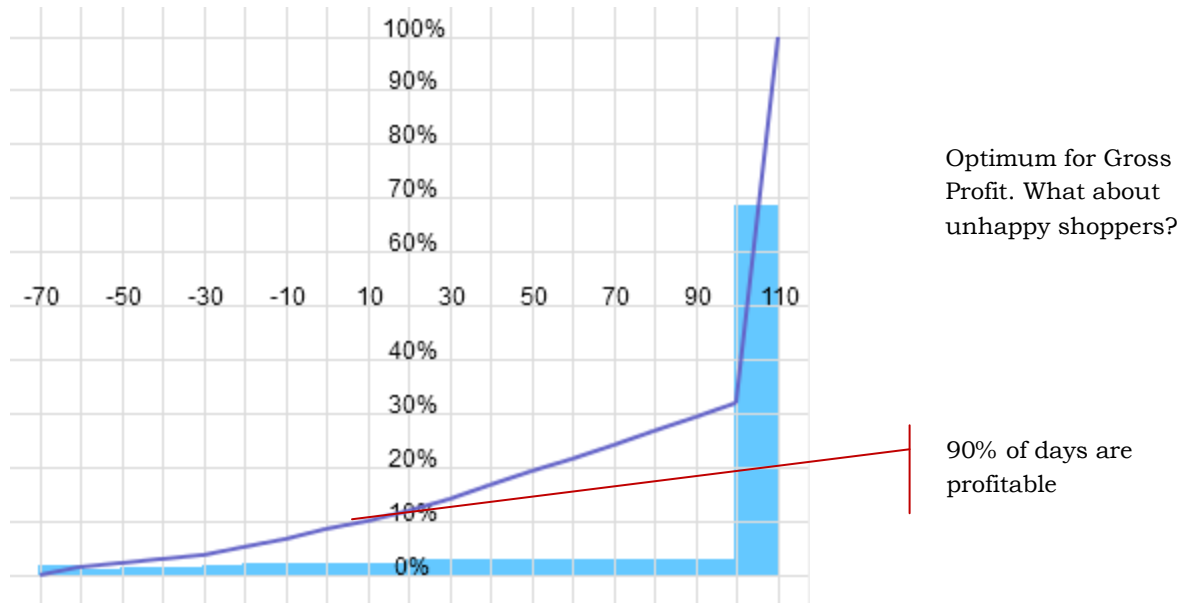
Let's look at preparing only 22 birds:



I picked 22 because that's your "can't lose" number. Your expected gross profit is \$64.12, the minimum is break-even and the cap is \$66. A nice, safe, conservative plan.

You could try a bunch of different stock levels and you'd find that the optimum is 34 pullets prepared per period. It has the highest expected gross profit: \$78.47. How does that fit with your guess?

Here's the SIP:



Aside from the maximum expected gross profit, what's notable is a less than 10% chance of a loss, and a 68% chance of making over \$100 on any given day (the flip side of a 32% chance of making \$100 or less).

That was a really simple SIP math exercise using calculations with only one SIP and a scalar (a single number with no structure).

There's an Excel workbook at <http://smpro.ca/sipmath/CalcUA01.xlsx> with the model for this example in it.

CALCULATING WITH SIPS AND SIPS

SIP math uses the simplest form of array math – no matrix inversions, no outer or inner products – just element-by-element operations.

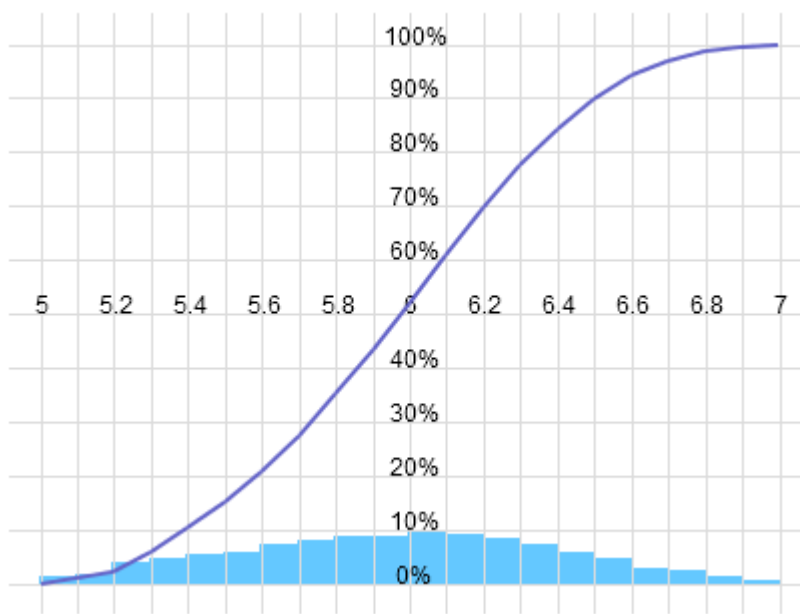
To add two SIPs, you add the first sample of one to the first sample of the other, then the second sample to the second sample, etc. For each trial i the result SIP sample is

$$\text{result}(i) = \text{varA}(i) + \text{varB}(i)$$

This also calculates the Joint Probability Distribution – but we don't care.

This holds for any operation that's valid for two scalars.

Let's say the cost of your chickens varies. Instead of a nice neat \$6, your unit cost fluctuates between \$5 and \$7. It needs a SIP, so you go through your accounts and you get a year's worth of samples (365) of past costs. Here's the SIP:



Because we're simulating 1,000 trials, we need to convert the 365 samples to 1,000 samples. The easiest way to do this is called *re-sampling*. We select samples from the costs SIP at random, doing it 1,000 times to compose the SIP we'll use in the model. This *random permutation* is pretty much the only use we have for a random number generator.

Look at the companion workbook at <http://smpro.ca/sipmath/CalcUB01.xlsx> on the **Library** sheet.

The new formula for UnitCost in trial i is

$$\text{UnitCost}(i) = \text{CostsA}(\text{Sampler}(i))$$

Resample to change size.

Where

CostsA is the original 365-sample SIP, and

Sampler is a vector of 1000 random integers between 1 and 365.

The formula for each cell in *Sampler* was

$$=1+\text{FLOOR}(\text{RAND()}*365,1)$$

B	C	D	E	F
<i>PM_Trials</i>	<i>Demand</i>	<i>CostsA</i>	<i>Sampler</i>	<i>UnitCost</i>
1000	20	6.25	289	6.31
	37	6.38	331	5.61
	30	6.32	113	5.45
	62	5.84	210	6.45
	46	6.39	147	5.81
	108	6	275	6.32
	72	6.57	263	5.96
	17	5.34	64	6.35
	27	6.23	268	6.06
	50	6.43	38	6.34
	73	6.66	227	5.88
	45	6.22	212	5.22

When this is all done in the workbook, we select the whole mess and Copy | Paste | Special | Values to get rid of the formulas and freeze the *Sampler* index and SIPs forever. We don't need to keep recalculating them. One of the main benefits of this approach is leaving an audit trail – not just of the formulas but of the intermediate results as well.

It also handles an important SIP management issue.

A SIP has two important, independent, properties: shape and order.

A SIP's *shape* is its sample values without reference to the order they're in. The shape gives a SIP its statistical attributes. The percentile curve and histogram are completely determined by the shape.

A SIP has Shape and Order.

A SIP's *order* is the sequence the values are in, and it needs to be managed when we're operating on multiple SIPs.

When we're dealing with unrelated values like the cost of raw chickens and demand for the cooked version, we want to make sure we get a full range of different combinations of the two, just as it happens in the real world. Random permutation – shuffling – of the SIP vectors achieves that goal.

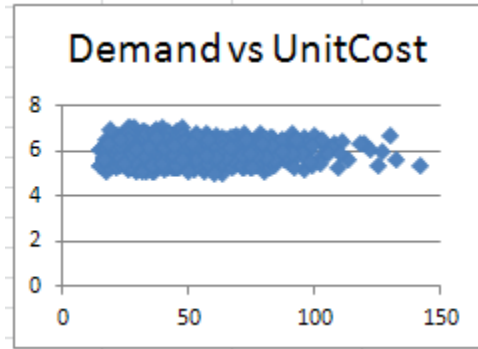
There's a hitch.

Let's say a model includes the uncertain cost of orange juice. You can buy juice or, if the price is right, buy oranges and juice them yourself. The price of oranges and the price of juice are both SIPs, and the formula for your cost is the minimum of the costs associated with either buying juice or making juice.

What's different is that the price of oranges and the price of orange juice are entangled. They'll tend to rise and fall together. Any particular pair of samples needs to preserve the relationship found in the market. We can't just shuffle them independently; we'd get pairs of values that would be very unlikely in the real world and that would bias the results of the simulation. The two SIPs comprise a *SLURP* – a Stochastic Library Unit with Relationships Preserved.

Entangled, not correlated.
Correlation index is too weak a measure.

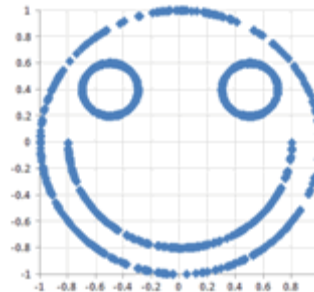
The only reliable way to check for suspected entanglement is to look at a scatter chart of the two SIPs, like this:



If, as in this case, you get a horizontal or vertical blur, entanglement is not an issue. If you get an angled blur or an interesting pattern, you can assume entanglement.

In more abstract terms, if reducing the uncertainty in one of the SIPs reduces the uncertainty in the other, they're entangled. Maybe like Sam Savage's Happy Face.

The happy face is a scatterplot of two SIPs whose index of correlation is a negligible .008. Clearly, the two SIPs are entangled even though they're theoretically not correlated.



Flaw of Averages
p.107

To get effective input mixing with other SIPs in the model, the SIPs in a SLURP still need to be permuted. A SLURP's *coherence* is maintained by permuting all of its constituent SIPs with the same permutation index. The scatter plots after permutation should look exactly the same as before the permutation.

So the coherence rule is that *independent SIPs and SLURPs are permuted with different permutation indexes while entangled SIPs in SLURPs are permuted with the same permutation index.*

Explicitly including the permutation indexes with the model data makes both audit and debug a whole lot easier. That's why the model library includes a column for the *Sampler* permutation index.

Once the input SIPs are taken care of, all that's left is to step through them, running the model for each trial. In the workbook, we're using the SIPmath.xlam add-in to build the model, and a novel use of Excel's Data Tables and INDEX() function to run it. I won't explain it here, but refer you to the [SIPmath Tutorial](#) at SIPmath.org.

HOW INTERESTING CAN IT GET?

If your model looks like a network (e.g. project plan, process model), the “whatever you can do with a scalar” still applies.

We've already used *minimum*. For a project plan, you can use *maximum* to select the latest finish for a bunch of tasks converging on a milestone.

You can also use relational operations. Let's say you have an identified risk and the event has a 25% chance of happening. If it does, it will add \$100,000 to the cost of a particular task.

You'll resist the temptation to just add the average \$25,000 to the task cost because that's not what happens in the real world. It's one project, not a million transactions, so the average is invalid. In each trial, it's \$100,000 or nothing.

Risk = Probability x Impact
is wrong.

It's possible that downstream events would be triggered by the \$100,000 while \$25,000 would fly under the radar. Also, looking at the range of possible project costs, the high numbers would be \$75,000 low, and the low numbers would be \$25,000 high. So don't use *Probability x Impact*. Ever.

All you need is a SIP of ones and zeros set up so that 25% of the samples are ones – let's call it riskP.

Then, for trial i

```
taskCost(i) =  
taskBaseCost(i) + if( riskP(i), 100000, 0)
```

If the riskP element is 1, you include the \$100,000 in that trial, otherwise you don't.

It even works if the risky cost is uncertain, say between \$75,000 and \$125,000 with an interesting shape. (This is a common case that risk analysts try to ignore.) Say you have it in a SIP named riskC:

```
taskCost(i) =  
taskBaseCost(i) + if(riskP(i), riskC(i), 0)
```

The logical functions AND(), OR(), XOR(), etc. are also valid so if, for example, you have a cost that depends on at least one of two uncertain events happening:

```
addC(i) = if( OR(evA(i),evB(i)), xCost(i), 0)
```

The bottom line is this: whatever model you could build if all your values were simple constants, you can quickly turn it into a simulation and run thousands of trials with different combinations of inputs – producing output SIPs that will give you insight into the range of possibilities and their likelihood – instead of a single number with questionable parentage and utility.

COMPOSING SIPS

Given a variable whose value is unknown, composing a SIP has these requirements:

- 1 Every sample's value is a possible value for the uncertain variable.

- 2 The range of possible values for the uncertain variable is well represented by the sample values.
- 3 For any range of values, the percentage of samples in the range is near the probability that the variable will take on a value in that range.
- 4 The SIP order meets whatever need for coherence applies.
- 5 The SIP minimum and maximum are close to the variable's minimum and maximum.

Rule 5 may be relaxed in the case of 'long tailed' variables. *How* and *when* is a risk management issue that we can't help with the math, so it's outside our scope. Estimating very low probabilities is very hard, so preparation usually trumps prediction where this is a factor.

Keep in mind you're looking for *improvement* over calculating with averages, correctly quantifying and reducing the uncertainty, not eliminating it.

Using historical observations, as we've done for all the above SIPs, is easiest and most reliable if you can do it.

If the observed data aren't directly applicable, some function may be. For example

- If the task is laying track for a light rail transit project, you can look at the final costs of a lot of deployed LRT projects and convert them to cost per mile of track, giving you a SIP you can use to scope your costs.
- The construction industry has a variety of sources for standard costs. Your history of actual/standard costs is a SIP that can be used to SIPify the scalar standard cost.

Beware the long tail.

There are other ways (e.g. Importance Sampling) to represent sample distributions that help with low probabilities, but they don't use SIPs or SIP math.

John Marc Thibault, *The Art of the Plan*, (2012)
Section 4.1: Outside View
for an LRT project.

- Software predictive metrics like function points can also be turned into work per point SIPs (instead of averages) with actual work per point history.

If good history isn't available, the final option is estimating. Read Douglas Hubbard's *How to Measure Anything* first.

An effective way to estimate a SIP is to reverse-engineer a histogram. Use your subject-matter expert to estimate the minimum and maximum values, and the relative probabilities of the values in between, to create a histogram. Then convert that histogram into a SIP. An easy way is to use [DistShaper](#) to shape the histogram and generate the SIP, which you can copy and paste into your Excel SIP library. (I'm working on an Excel version, so stay tuned.)

DistShaper reverse-engineers a histogram into a SIP.

You can also use [SDXL](#)'s `fromWxV()` function to generate a SIP given a list of bounding values and a corresponding list of weights. Set up two data ranges: one each for values and frequency. Have them act as the series for a vertical bar chart (values are the X-axis) and prime them with the value intervals and a starting set of frequencies. Create a chart using them.

Now adjust the frequencies until you're happy with the histogram. Use `fromWxV()` to compose a SIP of the desired size using the two data ranges. Copy the SIP values only into your SIP library so that the model doesn't need SDXL to run.

Finally, if you know your way around the statistics, and you believe your uncertain variable can be approximated by a famous-name probability distribution, you have a couple of options:

- 1 Use the relevant Excel inverse function in an array formula to generate the values to populate the SIP.

- 2 Use one of the predefined famous-name SIPs from SIPmath.org's SIP Library.

WHAT'S NEXT?

Explore the resources listed on the next page, go to sipmath.org, join up as a member, improve your models, and help us fix the Flaw of Averages wherever it raises it's ugly misleading head.

If you need help, or just want to talk, I can be reached at marc@smpro.ca or on [Google+](#).

RESOURCES

- Workbooks** smpro.ca/sipmath.
- Sipmath.org** SIP and Probability Management Headquarters.
- SIP math tools** *SIPmath.xlam* is an add-in for building Excel models applying SIP math techniques. The add-in is used just for building the model; it's not needed to run it. Get it from Sipmath.org.
- DistShaper** Shape a histogram and generate the corresponding SIP. Free Web app. smpro.ca/pjs/distshaper.
- SDXL** *Sample Distributions for Excel* takes a parallel approach to modeling with SIPs. It's a free, open source, and well-documented Excel add-in useful for developing large complex models. Get it at smpro.ca/SDXL.
- XLSim** An Excel add-in with lots of user interface bells and whistles, and partial automation of the model building process. Get it at xlsim.com.
- Blog** [The Art Of the Plan](#).
- Books** [The Flaw of Averages](#). The problem and the solution.
[The Art of the Plan](#) Print and Kindle editions.