

Copulas: when your inputs **move together**

A simulation that draws every input on its own quietly assumes they have nothing to do with each other. In a real business, volume, price and cost are linked. A copula lets you keep each input's own distribution and still make them move together, as much or as little as you believe they do.

Functions	<code>fx.RiskCopulaGauss</code> · <code>fx.RiskCopulaT</code> · <code>fx.RiskCholesky</code> · Clayton, Gumbel and Frank
In XL Edge	XL Edge tab > Productivity Tools > Stat Tools > Insert Monte Carlo Copula
Example	<i>XL Edge Example - Copula.xlsx</i> (same folder): a manufacturer's income statement, 10,000 trials. Every number in this guide is read from it.
Needs	Excel for Microsoft 365. The functions also work without XL Edge once they are in a workbook.

1 The idea in one paragraph

2 Using the ribbon

3 The example: assumptions

4 The calculation engine

5 Checking that the copula did its job

6 Results: what correlation changes

7 The charts

8 Choosing a copula

9 Practical advice

10 Argument reference

1 The idea in one paragraph

Every XL Edge distribution turns a column of **uniform random numbers** (values between 0 and 1) into draws from that distribution. A low uniform gives a low value and a high uniform a high one. Normally each input gets its own, unrelated uniforms. A copula hands out the uniforms instead, **already correlated**: one column per input, with the ones meant to rise together tending to be high together. Each input still has exactly the distribution you gave it, because the copula changes only *which* draws come together, not the draws themselves. That separation, the marginals on one side and the dependence on the other, is the whole idea, and it is a theorem (Sklar's): any joint distribution can be built this way.

TWO CATCHES, BOTH HANDLED IN THIS GUIDE

The ρ you type is the correlation of the *normal scores* behind the uniforms, not of the inputs themselves (section 5 shows how close they come). And the matrix must be internally consistent, which mathematicians call **positive definite**. A matrix filled in pair by pair can fail that, and the copula then returns #NUM! (section 9).

2 Using the ribbon

XL Edge tab > Productivity Tools > Stat Tools (arrow) > Insert Monte Carlo Copula

Menu item	XL Edge asks for	Result
Gaussian (Normal)	The correlation matrix (a square range, 1s on the diagonal)	Trials × d uniforms
Student t	The matrix, then the degrees of freedom (smaller = extremes cluster more)	Trials × d
Clayton, Gumbel, Frank	Theta, the number of variables, an optional Rotation	Trials × variables
Clayton / Frank (negative dependence)	Theta below 0; two variables that move in opposite directions	Trials × 2

If the workbook has no trial count yet, XL Edge asks for one and stores it as the name `MC_Trials`. It then asks where the block goes, tells you its size, refuses to overwrite anything, and gives the copula the next free `VarID`. Its closing message shows how to use a column:

```
the copula in E5 spills one column per input; feed column 1 to the first input's distribution  
=fx.RiskNormal(100, 15, CHOOSECOLS(E5#, 1))
```

Any distribution accepts the column in place of its `Trials` argument. Insert each input with **Insert Monte Carlo Distribution** as usual, then replace its `Trials` with `CHOOSECOLS(...#, j)`.

3 The example: assumptions

A medium-sized manufacturer with good economies of scale plans next year's income statement: about 80 million dollars of revenue at a 40% gross margin. Six inputs are uncertain. Tax is a flat 21% of pre-tax profit, with no benefit taken on a loss. Sheet *Assumptions* holds every input in a yellow cell, with a note beside each.

Uncertain inputs (sheet *Assumptions*)

Input	Distribution	Parameters	Why this shape
Sales volume (units)	PERT	330,000 / 400,000 / 430,000	Plant capacity caps the upside; a lost customer can cost a lot. So it leans left.
Sales price (\$)	Truncated normal	mean 200, SD 8, from 180 to 220	A contract floor and competitors' list prices bound it.
Cost per unit (\$)	Lognormal	mean 120, SD 7	Overruns run bigger than savings: skewed right.
Sales expense (% of revenue)	Beta, scaled	α 4, β 6, from 6.0% to 11.0% (mean 8.0%)	A bounded percentage whose long tail is overspending.
Other variable costs (% of revenue)	SPT metalog	P10 4.3%, P50 5.0%, P90 6.0%; bounds 3% to 9%	The controller's low / likely / high for freight, utilities and warranty.
G&A expense (\$)	Normal	mean 9,000,000, SD 250,000	Semi-fixed; it moves only with small variances to plan.

A TRAP WORTH KNOWING: THE LOGNORMAL'S PARAMETERS

`fx.RiskLogNormλ`, like @RISK's RiskLognorm2, takes the mean and SD of $LN(cost)$, not of cost. The sheet converts them: $\sigma = \sqrt{LN(1 + (SD/mean)^2)} = 0.0583$ and $\mu = LN(mean) - \sigma^2/2 = 4.7858$. Typing 120 and 7 straight into the function would simulate costs around e^{120} .

The dependence

Only volume, price and cost per unit go into the copula. The other three vary on their own. The matrix is a set of business beliefs, one per pair:

Pair	ρ	The belief
Volume and cost per unit	-0.60	Economies of scale: more units spread the fixed overhead, so cost per unit falls.
Volume and price	-0.50	The demand curve: extra volume is won with price.
Price and cost per unit	+0.40	When material costs rise, prices follow part of the way.

Only the three yellow cells above the diagonal are typed. The cells below mirror them, so the matrix is always symmetric, and a live check beside it (`fx.RiskCholeskyλ`) answers "Valid (positive definite)?" with **Yes**.

Three scenarios. The model runs three times side by side: **1 Independent** (the identity matrix, which means no correlation), **2 Gaussian copula** (the matrix) and **3 Student t copula** (the same matrix with 4 degrees of freedom, so extremes also tend to come together).

4 The calculation engine

Sheet *Simulation_Calculations* holds one spilled formula per column: the copula's three columns of uniforms, then the six inputs, then the nine income statement lines, one row per trial. Five names keep the formulas readable: `Iterations`, `SimSeed`, `CorrelMatrix`, `TDegFree` and `TaxRate`, all pointing at

Assumptions. Scenario 2 starts in column AA:

```
AA9: the copula, Iterations rows by 3 columns of correlated uniforms
=fx.RiskCopulaGaussλ(CorrelMatrix, Iterations, 1, SimSeed)
AD9:AF9: each copula column replaces the distribution's Trials argument
=fx.RiskPertλ(Assumptions!$D$14, Assumptions!$D$15, Assumptions!$D$16, CHOOSECOLS(AA9#, 1))
=fx.RiskTruncNormalλ(Assumptions!$D$20, Assumptions!$D$21, Assumptions!$D$22,
Assumptions!$D$23, CHOOSECOLS(AA9#, 2))
=fx.RiskLogNormalλ(Assumptions!$D$29, Assumptions!$D$30, CHOOSECOLS(AA9#, 3))
AJ9 onward: the income statement, e.g. revenue, then tax (no benefit on a loss) and net income
=AD9#*AE9#
=IF(AP9#>0, AP9#*TaxRate, 0)
=AP9#-AQ9#
```

Scenario 1 (column H) uses `IdentityMatrix` (α `=MUNIT(3)` on *Assumptions*) in place of `CorrelMatrix`; Scenario 3 (column AT) uses `=fx.RiskCopulaTλ(CorrelMatrix, TDegFree, Iterations, 1, SimSeed)`. The three inputs outside the copula are drawn once, in Scenario 1, with their own VarIDs (11, 12, 13):

```
=fx.RiskBetaλ(Assumptions!$D$34, Assumptions!$D$35, Assumptions!$D$36, Assumptions!$D$37,
Iterations, 11, SimSeed)
```

Scenarios 2 and 3 link to those columns (`=N9#`), so the draws are visibly the same.

SAME RANDOM NUMBERS, DIFFERENT DEPENDENCE

All three copulas use VarID 1 and the same Seed, so they start from the **same** normal random numbers. The identity matrix passes them through untouched, the Gaussian copula mixes them with the matrix, and the t copula mixes them the same way and then divides each trial by one shared random factor (from a fourth stream). Any difference between the scenarios is therefore the dependence, not the luck of the draw. Analysts call this *common random numbers*.

5 Checking that the copula did its job

Sheet *Outputs* ends with the checks, each shown as check, expected and difference, with a blank *Manual sign-off* cell for your ✓. A correlation estimated from 10,000 trials has a standard error of about 0.01, so differences within ± 0.03 pass.

Scenario 2 (Gaussian) and Scenario 3 (t) against the matrix

Pair	Target ρ	Normal-score correlation	Expected rank correlation	Achieved rank correlation	t-score correlation
Volume / Price	-0.50	-0.485	-0.483	-0.466	-0.477
Volume / Cost per unit	-0.60	-0.595	-0.582	-0.574	-0.577
Price / Cost per unit	0.40	0.388	0.385	0.370	0.375

- **Normal-score correlation**, `CORREL(NORM.S.INV( $u_i$ ), NORM.S.INV( $u_j$ ))`, undoes the copula's last step, so it should equal ρ . It does, within sampling error.

- **Rank (Spearman) correlation** is what survives into the inputs, whatever their distributions, because every distribution keeps the order of its uniforms. For a Gaussian copula it should be $6/\pi \cdot \text{ASIN}(\rho/2)$, a little smaller than ρ . The sheet computes it with `fx.RiskCorrelMatrixλ(..., , "Spearman")` on the simulated inputs.
- **t-score correlation** does the same for the t copula, with `T.INV`. It is noisier because the t copula's extremes are heavier, so allow about ± 0.05 .
- The independent scenario's largest normal-score correlation is 0.016, which is zero within sampling error.

Each input also matches its distribution. The theoretical mean (the median, for the metalog) sits within three standard errors of the simulated one: volume 393,217 against 393,333, cost per unit 119.90 against 120.00. The copula moved the draws around; it did not change them.

6 Results: what correlation changes

Net income, \$ millions (sheet *Outputs*)

	1 Independent	2 Gaussian	3 Student t
Mean	9.67	9.67	9.67
Standard deviation	3.17	2.62	2.64
P10 to P90	5.62 to 13.76	6.34 to 13.04	6.46 to 12.91
Range compared with Independent	100%	82%	79%
P1 (a 1-in-100 year)	2.19	3.56	3.18
Worst of the 10,000 trials	-2.08	0.83	-5.80
"Perfect storm": volume in its lowest 5% <i>and</i> cost in its highest 5%	0.24%	1.54%	2.09%
... times as likely as independence implies (0.25%)	1.0	6.2	8.4
Probability of falling short of the plan (9.95)	53.6%	54.7%	54.8%

1. Here, correlation narrows the range. The mean hardly moves, because a copula leaves every input's own distribution alone. But the P10 to P90 range shrinks to 82% of the independent one. The beliefs in the matrix act as a *natural hedge*. A big-volume year also brings lower prices (bad) and lower unit costs (good), and when costs rise, prices partly follow. Each pair of effects partly cancels. An independent model misses the cancelling and **overstates this company's risk**, which matters if the range drives a budget reserve, a covenant cushion or a go / no-go decision.

2. But the bad corner gets more likely. Correlation is not only "less risk". The perfect storm, a weak volume year with high unit costs, happens in 1.54% of Gaussian trials, 6.2 times what independence implies. Under the t copula it is 2.09%. That is the economies-of-scale belief working in reverse: low volume *causes* high unit cost. Such years average 5.71 million of net income against a mean of 9.67.

3. The tail is where Gaussian and t differ. The two copulas share the same matrix and agree almost everywhere: same mean, nearly the same range. In the far left tail, the t copula's 1-in-100 year is 3.18 million against 3.56, and its single worst trial is a loss of 5.80 million. The Gaussian's worst trial still earns

0.83. If a bad year is what you are planning for, the choice of copula matters even when the correlations are identical.

4. The plan is optimistic, whatever the dependence. The single-point plan, built from each input's most likely or average value, gives net income of 9.95 million. The simulated mean is 9.67, and 54.7% of trials fall short. Volume's most likely value (400,000) sits above its mean (393,217) because the distribution leans left, so a plan built on most-likely values starts high. This is the *flaw of averages*.

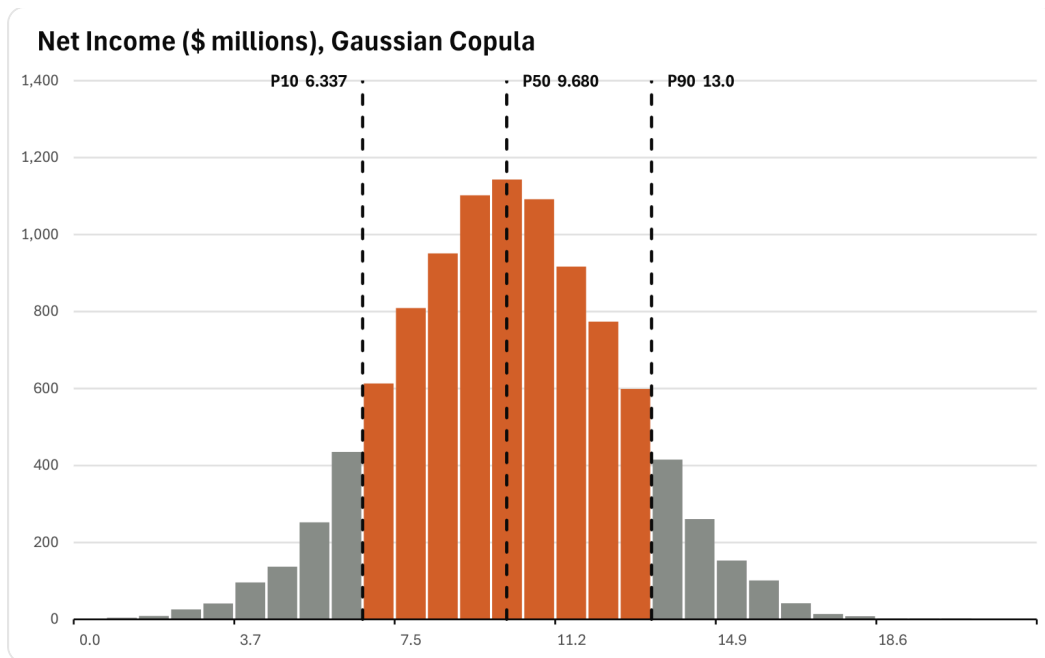
Scenario 2 income statement: plan against simulation, \$ millions

Line	Plan	Mean	P10	P90	Mean – Plan
Revenue	80.00	78.60	74.14	82.97	-1.40
Cost of goods sold	48.00	47.07	44.21	49.98	-0.93
Gross profit	32.00	31.53	26.98	36.10	-0.47
Sales expense	6.40	6.29	5.48	7.15	-0.11
Other variable costs	4.00	4.00	3.35	4.75	0.00
G&A expense	9.00	9.00	8.69	9.31	0.00
Pre-tax profit	12.60	12.24	8.02	16.50	-0.36
Income tax (21%)	2.65	2.57	1.68	3.47	-0.08
Net income	9.95	9.67	6.34	13.04	-0.28

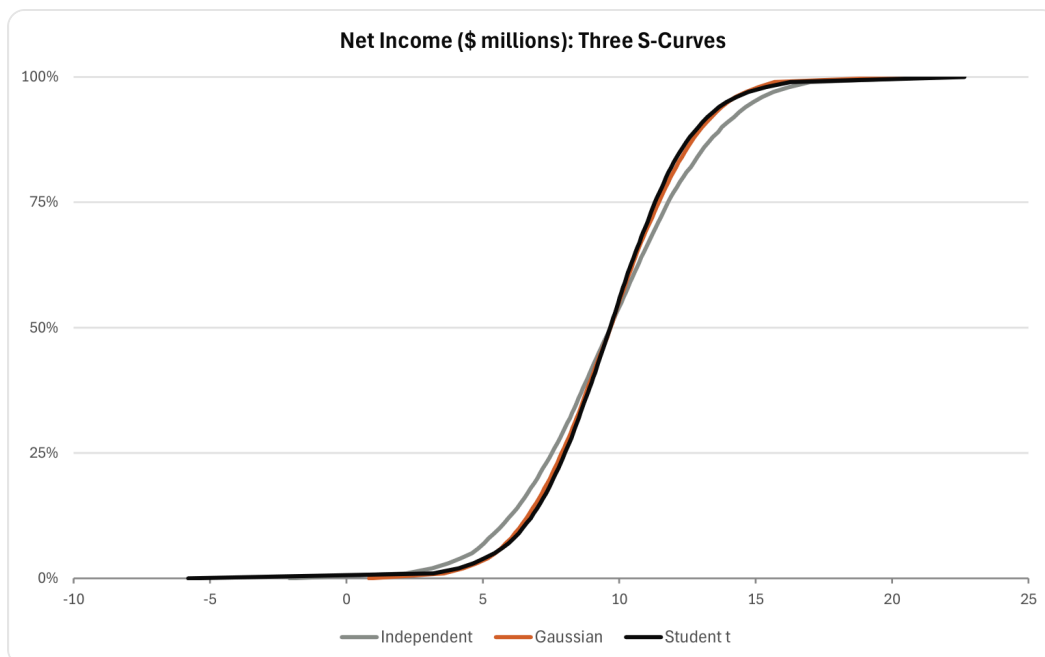
The statement ties: gross profit less expenses, tax and net income is 0.00 in the plan column and 0.00 in the mean column (averages add up line by line, though percentiles do not). The simulated gross margin is 40.1% and the pre-tax margin 15.6%.

7 The charts

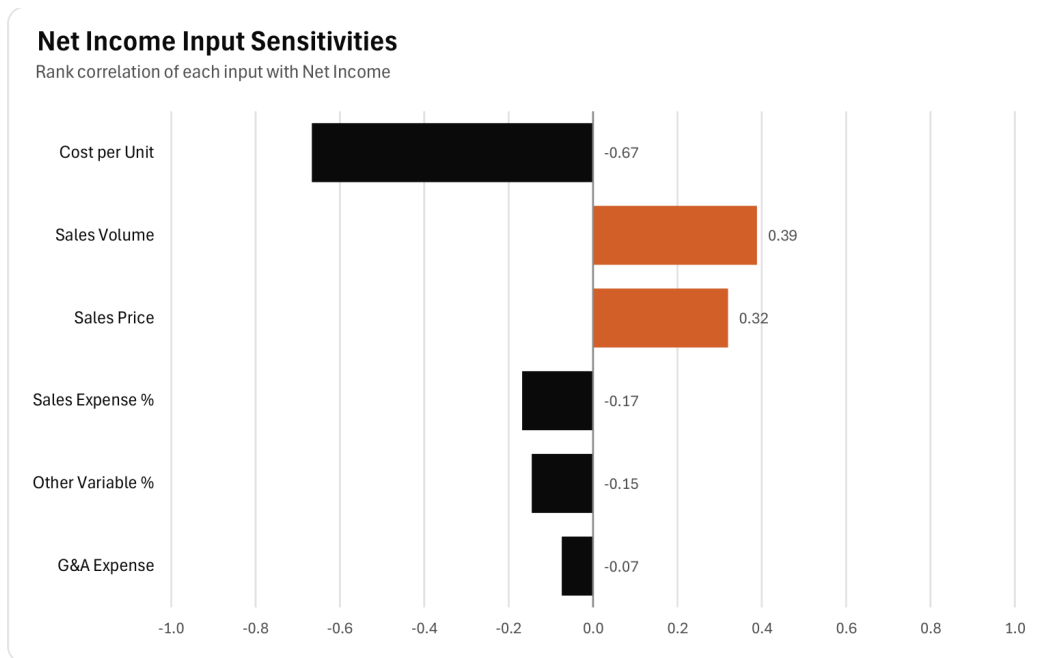
Sheet *Charts* keeps the charts apart from the engine, each fed by a block of chart data to its right. The first and third come from XL Edge's chart menu (**Insert Monte Carlo Chart**); the others are ordinary Excel charts over `fx.RiskChartCDFA` and the copula's uniforms.



Net income, Gaussian copula: `fx.RiskChartHistλ` on net income ÷ 1,000,000, drawn by *Outcome Histogram* (P10 / P50 / P90). Orange is the middle 80% of trials.



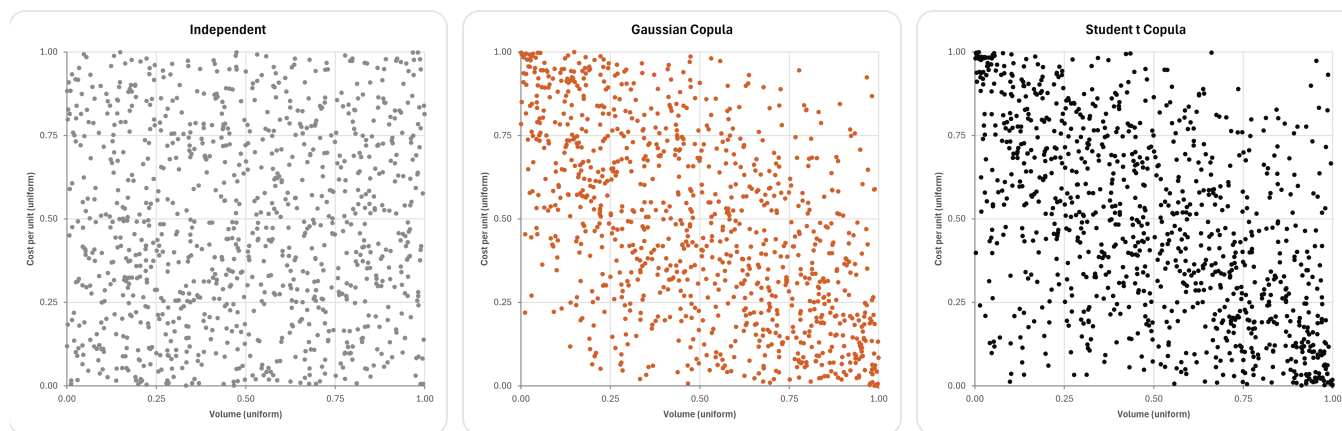
Three S-curves. Read across at any probability to see that year's net income. The Gaussian (orange) and t (black) curves are steeper than the independent one (gray), which is the narrower range. On the far left, the black curve reaches lowest: the t copula's worst years.



What drives net income (Gaussian): `fx.RiskTornado` with Method "rank", drawn by *Tornado (Sensitivity)*. Cost per Unit leads at -0.67 , then Sales Volume at 0.39 and Sales Price at 0.32 .

READ A TORNADO OF CORRELATED INPUTS WITH CARE

When inputs move together, they share credit. Cost per unit ranks first partly because it carries volume's effect with it: in a big-volume year, unit cost is low *and* volume is high. A tornado shows association, not a clean split of cause. With independent inputs (Scenario 1) the bars would rank each input on its own.



The copula itself: the uniforms for volume (across) and cost per unit (up), first 1,000 trials. Independent is an even cloud. The Gaussian copula tilts it, so high volume comes with low cost. The t copula tilts it the same way and also crowds the two corners: the perfect storm sits at top left.

8 Choosing a copula

Copula	Dependence	Use it when
Gaussian	A full matrix; no extra tendency for extremes to coincide	The default. Any number of inputs, any mix of positive and negative pairs.
Student t	The same matrix, plus extremes that come together in <i>both</i> tails; fewer degrees of freedom means more	Bad (and good) years cluster: credit, markets, supply shocks. Above about 30 degrees of freedom it is the Gaussian.
Clayton	One θ for all pairs; variables crash together more than they boom together (Kendall's $\tau = \theta/(\theta+2)$)	Joint downside, e.g. losses across business lines in a downturn. Rotation 180 flips it to the upside.
Gumbel	One $\theta \geq 1$; variables boom together more than they crash together ($\tau = 1 - 1/\theta$)	Joint upside or joint large claims. $\theta = 1$ is independence.
Frank	One θ ; symmetric, with lighter tails than the Gaussian	Moderate, even dependence with no tail clustering.

Why the example uses Gaussian and t. Clayton, Gumbel and Frank have *one* parameter for every pair, so all pairs get the same strength and sign. Negative dependence works for two variables only (Rotation 90 or 270, or the two "negative dependence" items). This model needs three different correlations, two of them negative, and only a matrix copula can give it that.

9 Practical advice

- **Getting the numbers.** From history, use `fx.RiskCorrelMatrixλ(data, , "Spearman", , , FALSE)` on matched periods, then convert each rank correlation to the copula's ρ with $\rho = 2 \cdot \text{SIN}(\pi \cdot \rho_s / 6)$. Without data, ask in ranks: "in a bottom-quartile volume year, how often is unit cost in its top quartile?" People judge ranks far better than correlation coefficients.
- **A matrix can be impossible.** Volume and price at +0.9, volume and cost at +0.9, but price and cost at -0.9 cannot all be true at once. Such a matrix is not positive definite, and the copula returns `#NUM!`. Check with `=ISNUMBER(SUM(fx.RiskCholeskyλ(matrix)))`, and fix it by softening the pair you believe least.
- **Model the mechanism when you know it.** A copula captures "tends to move together". If you know *how*, for example that unit cost is fixed overhead ÷ volume plus a variable cost, a formula is better than a correlation. Use the copula for the links you can only describe statistically.
- **Sampling error.** With 10,000 trials the mean is good to about ± 0.08 million (3 standard errors), but a P1 rests on just 100 trials and the single worst trial on one. Use more trials before quoting far tails, and compare scenarios on the same random numbers, as here.
- **VarIDs.** A copula uses its VarID for all its columns, on streams that never collide with ordinary distributions. Two copulas meant to be *independent* of each other need different VarIDs; the three scenarios here share VarID 1 on purpose.
- **Pass the whole column.** `CH00SEC0LS(AA9#, 1)` passes all the trials. A single cell would be read as a trial count, not as a uniform.

10 Argument reference

`fx.RiskCopulaGaussλ(Correlation, [Trials], [VarID], [Seed])`

Argument	Default	Meaning
Correlation	required	d × d matrix: symmetric, 1s on the diagonal, positive definite. ρ is the correlation of the normal scores.
Trials	10000	Rows of output. Output: Trials × d uniforms in (0, 1).
VarID, Seed	1, 0	Select the random streams (one per column). Change either for new draws.

`fx.RiskCopulaTλ(Correlation, DegFree, [Trials], [VarID], [Seed])`

Argument	Default	Meaning
DegFree	required	Degrees of freedom, > 0 (fractions allowed). Fewer = stronger tail dependence.
Other arguments as for the Gaussian copula. Uses d + 1 streams; the first d are the Gaussian copula's.		

`fx.RiskCopulaClaytonλ, ...Gumbelλ, ...Frankλ(Theta, [Dimensions], [Trials], [VarID], [Seed], [Rotation])`

Argument	Default	Meaning
Theta	required	Strength. Clayton > 0, Gumbel ≥ 1, Frank > 0. For two variables only, Clayton −1 to 0 and Frank below 0 give negative dependence.
Dimensions	2	Number of variables (columns).
Rotation	0	0 or 180 for any number of variables; 90 or 270 (negative dependence) for two.

`fx.RiskCholeskyλ(Matrix)`

Argument	Default	Meaning
Matrix	required	Square, symmetric. Returns the lower-triangular factor; <code>#NUM!</code> if not positive definite, <code>#VALUE!</code> if not symmetric.

The distributions in the example (each takes a copula column as Trials)

Function	Parameters	Notes
<code>fx.RiskPertλ</code>	Min, Mode, Max	Smooth three-point estimate.
<code>fx.RiskTruncNormalλ</code>	Mean, StDev, [Min], [Max]	A normal confined to a range.
<code>fx.RiskLogNormalλ</code>	Mean, StDev of LN(x)	Convert from the mean and SD of x first (section 3).
<code>fx.RiskBetaλ</code>	Shape1, Shape2, [A], [B]	A and B rescale it to a range.
<code>fx.RiskSPTMetalogλ</code>	Low, Median, High, [LowerBound], [UpperBound], [Alpha]	Three quantiles (P10 / P50 / P90 by default).
<code>fx.RiskNormalλ</code>	Mean, StDev	
Each is followed by [Trials], [VarID], [Seed]. Pass a copula column as Trials to correlate it; VarID and Seed are then ignored.		

References: Sklar (1959); Nelsen (2006), *An Introduction to Copulas*; McNeil, Frey and Embrechts (2015), *Quantitative Risk Management*, ch. 7; Savage (2009), *The Flaw of Averages*; Keelin (2016), *Decision Analysis* 13(4) (metalogs). The example's assumptions are illustrative, not any real company's. XL Edge and the fx.Risk... functions: © 2026 W Phillips, edgewisedata.com, MIT License.