

Quantitative Risk Management

Homework for Week 6

John Dodson

October 18, 2021

Problems

Solutions to these problems are due at the beginning of our last session, which is 5:30 PM on Monday, October 25. Please turn in your solutions to the TA. Include your U of Minn. student identification number on your submission to facilitate recording marks in the Canvas learning management system. Also include the names of any classmates you consulted with in developing your solutions.

This problem is about tail dependence and the residual risk of hedged positions.

Consider the expected shortfall on a portfolio with loss

$$L = e^{X_1} - e^{X_2}$$

where $X = (X_1, X_2)'$ is a random variable in \mathbb{R}^2 representing the assets' log-returns.

Even if we had formulaic characterizations (e.g. parametric density functions) for the log-returns, we generally cannot translate these into formulaic characterizations for the portfolio loss. So we can at best estimate risk metrics based on expansions or simulations. For this problem, we will use simulation. That means we need to generate (pseudo-)random variates for X_1 and X_2 and assemble them into variates for the loss to create an empirical distribution. Risk metrics can be estimated from this empirical distribution.

Fix the marginal distributions as particular members of the location-scale family of the symmetric NRIG with shape parameter $g = 2$,

$$\frac{X_i - 0.002}{0.02} \sim \text{NRIG}(2.)$$

for $i = 1, 2$, and the concordance in terms of Kendall's tau or pseudo-correlation ρ ,

$$\tau = \frac{2}{\pi} \sin^{-1} \rho = 0.8$$

Consider two different copulas: the Gaussian copula and the Student's- t_4 copula.

Draws from a normal copula can be generated by drawing from a bivariate normal¹ and applying the marginal distribution function to get results in $(0, 1) \times (0, 1)$.

Draws from a Student's- t_4 copula can be generated by drawing from a bivariate Student's- t_4 and applying the marginal distribution function².

¹If $(X, Y)'$ are i.i.d. mean-zero normal, then $(X, \theta X + \sqrt{1 - \theta^2}Y)'$ for $-1 \leq \theta \leq 1$ are also bivariate normal with the same marginals.

²The distribution function of a (univariate) Student's- t_4 r.v. X is $P(X < x) = \frac{1}{2} \left(1 + \frac{x(6+x^2)}{(4+x^2)^{3/2}} \right)$.

1. Estimate the 99% value-at-risk and expected shortfall for this portfolio under the Gaussian copula. **(4 points)**
2. Estimate the 99% value-at-risk and expected shortfall for this portfolio under the Student's- t_4 copula. **(6 points)**

Hints

One way of generating bivariate Student's- t_4 variates is to divide bivariate normal variates by the square root of the average of four squared independent normals variates.

Since it is easy to make mistakes in generating copula draws, it is a good idea to verify your work by checking that the draws are marginally uniform in $(0, 1)$. For example, their sample means should be near $1/2$, and their sample variances should be near $1/12$.

In order to convert copula draws into log-returns, you will need to evaluate the quantile function (the inverse CDF) of the NRIIG. Unfortunately, this is expensive to compute. If we need to construct *both* the distribution and the inverse numerically, using the Newton-Raphson method for the inverse is an obvious technique to consider because we know that the distribution is monotonic and we have ready access to its derivative (the density).

The general updating scheme to approximate a root $x^* : h(x^*) = 0$ is $x_{i+1} \leftarrow x_i - h(x_i)/h'(x_i)$. In our case, $h(x) = \int_{-\infty}^x f(x') dx' - p$ for probability density $f(\cdot)$; so, for a symmetric density, the scheme is $x_1 = 0$ and

$$x_{i+1} \leftarrow x_i - \left(\frac{1}{2} - p + \int_0^{x_i} f(x') dx' \right) / f(x_i)$$

In general the Newton-Raphson method has quadratic convergence. The error seems to approach machine precision in four or five iterations.

Solution

Simulation is required to analyze the properties of the loss here, so it is primarily an exercise in scientific computing. To start, I defined the main constants.

```
"returns mean"
```

```
 $\mu = 0.002$ 
```

```
"returns deviation"
```

```
 $\sigma = 0.02$ 
```

```
"returns NRIIG shape"
```

```
 $g = 2.$ 
```

```
"returns Kendall correlation"
```

```
 $\tau = 0.8$ 
```

```
"returns pseudo-correlation"
```

```
 $\rho = \sin(\pi/2*\tau)$ 
```

```
"simulation size"
```

```
N = 100_000
```

```
"empirical VaR boundary"
```

```
N99 = floor(Int,0.99N)
```

I then created bivariate normal variates with the given correlation ($\rho \approx 0.951$), by combining independent ones, and Gaussian copula draws by applying the marginal distribution function to each.

```
"correlated standard normal variates"
```

```
z = randn(N,2)*[1 ρ;0 sqrt(1-ρ^2)]
```

```
"gaussian copula draws"
```

```
u_gauss = erfc.(z/-sqrt(2))/2
```

Note here in the expression for z that Julia (like Matlab) treats multiplication of arrays as matrix multiplication (contraction on inner indices). Here, I am multiplying on the right by what amounts to the (upper) Cholesky decomposition of the correlation matrix:

$$\begin{pmatrix} 1 & \rho \\ 0 & \sqrt{1-\rho^2} \end{pmatrix}' \begin{pmatrix} 1 & \rho \\ 0 & \sqrt{1-\rho^2} \end{pmatrix} = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$$

As noted in the hint, it is always a good idea at this stage to apply some basic quality assurance tests. I checked the ranges, means, and variances of the margins, and got good agreement with the expected results. I also checked the sample rank correlation, and got $\hat{\tau} \approx 0.799$, which is in good agreement with the design.

For the t_4 copula, I started by converting my bivariate normal sample to a bivariate Student's- t sample, by dividing each by an independent χ_4 (constructed by averaging the squares of four independent normals).

```
"correlated Student's-t4 variates"
```

```
t4 = z./sqrt.(mean(randn(N,4).^2,dims=2))
```

The margins of this are also Student's- t_4 ; so to extract copula draws apply the Student's- t_4 distribution function:

```
"t4 distribution function"
```

```
t4_dist = x -> (1+(x*(6+x^2))/((4+x^2)^(3/2)))/2
```

```
"t4 copula draws"
```

```
u_t4 = t4_dist.(t4)
```

Again, I checked the results for the expected ranges, means, marginal variances, and the rank correlation. Samples from both copulas are plotted in Figure 1.

Now that I have the copula draws, I need to convert them into returns by applying the NRIQ quantile. This is the most algorithmically complex part of the exercise, because there is no simple formula for the inverse (cumulative) distribution of an NRIQ.

We discussed this challenge as part of this week's case, and I am borrowing the implementation from there.

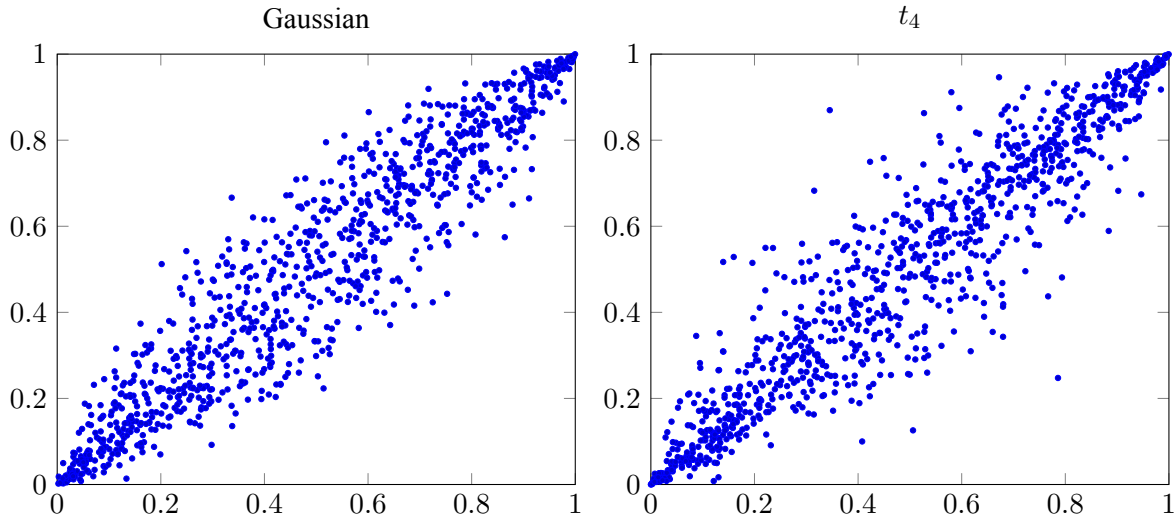


Figure 1: Samples from the two copulas for the assignment

"standard NRIX density"

```
NRIG_dens = (x,g) -> 1/π*exp(g)*sqrt(1+g)*besselk(0,sqrt(g^2+(1+g)*x^2))
```

"standard NRIX quantile"

```
function NRIG_quan(p,g)
# Newton-Raphson method to invert the standard NRIX distribution function
  x1 = 0. # initial guess
  x2 = 0.
  for n = 1:20
    f1 = NRIG_dens(x1,g)
    x2 = x1 - (1/2 - p + quadgk(x -> NRIG_dens(x,g), 0., x1)[1])/f1
    if abs(x2 - x1) < 1.E-12
      break
    end
    x1 = x2
  end
  return x2
end
```

Note that my version of QuadGK.quadgk returns two values: an estimate of the integral and an upper bound on the absolute error of the estimate (which I am ignoring here).

Applying the NRIX quantile is by far the most computationally intense part of the exercise. On my MacBook Air the following two lines each took about 15 seconds to run, compared to about 1 millisecond for the same calculation with a standard normal quantile function instead.

```
ret_gauss = μ. + σ * NRIG_quan.(u_gauss, g)
ret_t4 = μ. + σ * NRIG_quan.(u_t4, g)
```

The loss scenarios are therefore

```
L_gauss = sort(exp.(ret_gauss)*[1;-1])
L_t4 = sort(exp.(ret_t4)*[1;-1])
```

Finally, we can treat the sorted scenarios as an empirical distribution function and evaluate the risk metrics.

```
VaR_gauss = L_gauss[N99]
VaR_t4 = L_t4[N99]
ES_gauss = mean(L_gauss[N99:end])
ES_t4 = mean(L_t4[N99:end])
```

Results are presented below.

	Gaussian	t_4
VaR _{0.99}	0.0154	0.0166
ES _{0.99}	0.0183	0.0219

Table 1: Risk metrics for the hedged portfolio

Discussion

It is interesting that tail dependence in this setting has more of an effect on expected shortfall than it does on value-at-risk. We could summarize the results by concluding that the probability of experiencing an extreme loss, say greater than about 1.6%, does not depend very much on the model for tail dependence. But the *magnitude* of that loss does: a reasonably large tail dependence (provided by the t_4 copula in this example) causes extreme losses about 20% larger on average than under no tail dependence.

You might imagine that tail dependence would have the opposite effect: that a contagion would help to keep the long and the short aligned. But, that is not what we see. Tail dependence actually makes the hedge perform worse, not better—at least under a coherent risk metric.

It is perhaps *more* interesting to compare the risk of the hedged position to that of the unhedged position. Using the same scenarios, I got that the 99% expected shortfall on the bare long (whose return is X_2) is about 5.9%. A position with return X_1 would seem to be an excellent candidate for a hedge: The correlation is 0.95 and the volatility is identical. But the hedged portfolio still has expected shortfall somewhere between 1.8% and 2.2%. This apparently excellent hedge has only eliminated about two-thirds of the risk. And worse: the new risk, which we call “basis risk”, is difficult to measure³, difficult to explain, and impossible to manage.

³Any estimate of tail dependence present in a real-world sample must face the challenge that tail events are, by definition, rare.