

Practicing Copula Theory & Implementation
(working paper)

Eddie Shim

August 31, 2016

1 Introduction

In this exercise I will create a t -copula with the P&L distributions of Toyota and Tesla then calculate a 99.97% VaR and 2% Capital Allocation Tail then conduct a sensitivity test on the copula parameters. In this hypothetical, assume I am working with a portfolio consistent of half Toyota stock and half Tesla stock. This exercise serves as practice for me to code in Matlab, typesetting in LaTeX, and implementing copula theory (use of KDE and Monte Carlo simulation).

Describing Copula Theory: Intuitively a copula can be thought of as the information missing from individual marginals to complete a joint distribution (ie: "joint = copula + marginals") [1]. In this instance, we have two univariate probability distributions (Toyota and Tesla P&L) which we can infer exhibit some sort of correlation, as both belong in the auto industry. By calculating the copula dependency between these two distributions, we can join them into one joint distribution then pull separate marginal distributions from this joint distribution which strips the interdependencies between them.

Mathematically, a bivariate copula function is a function that links univariate marginals to their full bivariate joint distribution. Let C represent the copula function and U_1, U_2 represent uniform random variables in the following [2]:

$$C(u_1, u_2, \rho) = P(U_1 \leq u_1, U_2 \leq u_2) \quad (1)$$

let random variable $F_1(x_1)$ represent Toyota's P&L univariate marginal distribution and $F_2(x_2)$ Tesla's. The function

$$C(F_1(x_1), F_2(x_2)) = F(x_1, x_2) \quad (2)$$

uses the copula function C to create a bivariate joint distribution function with univariate marginal distributions $F_1(x_1)$ and $F_2(x_2)$. It follows that:

$$\begin{aligned} C(F_1(x_1), F_2(x_2)) &= P(U_1 \leq F_1(x_1), U_2 \leq F_2(x_2)) \\ &= P(F_1^{-1}(U_1), F_2^{-1}(U_2)) \\ &= P(X_1 \leq x_1, X_2 \leq x_2) \\ &= F(x_1, x_2) \end{aligned}$$

Sklar's Theorem established the converse, proving that from a bivariate joint distribution function, $\exists!$ copula function C such that you can pull the univariate marginal distributions:

$$F(x_1, x_2) = C(F_1(x_1), F_2(x_2)) \quad (3)$$

As a high level overview, here are the steps for building this model:

1. Get data of Toyota and Tesla P&L distributions X, Y .
2. Transform X, Y into copula scale variables U, V . This is achieved by feeding the random variable X through its own CDF:

$$\begin{aligned} U &= F_X(X) \\ U &\sim U_{[0,1]} \end{aligned}$$

where U is uniform on the unit interval. The CDF is obtained using nonparametric KDE estimation.

3. Estimate t -copula parameters ρ and ν from the transformed data using Maximum Likelihood Estimation.
4. Generate random samples from t -copula.
5. Transform the random samples back into the original scale of the data using KDE estimation of an inverse CDF. This gives us the univariate marginal distributions x_1, y_1 .
6. Calculate VaR and Expected Shortfall from the transformed random sample.

2 Data

Data was taken from Yahoo Finance, spanning daily from August 17, 2015 to August 11, 2016. This accounts for 250 observations. To obtain the P&L, simply difference the stock price data. Using a Kernel Density Estimator, I plotted the smoothed histograms of both P&L distributions. As observed in the below plots (Fig 1 & 2), both P&L distributions resemble a vague normal shape though leptokurtosis and fat tails are present.

```
%% DATA LOAD
filename_toyota = 'toyota_raw.csv';
filename_tesla = 'tesla_raw.csv';

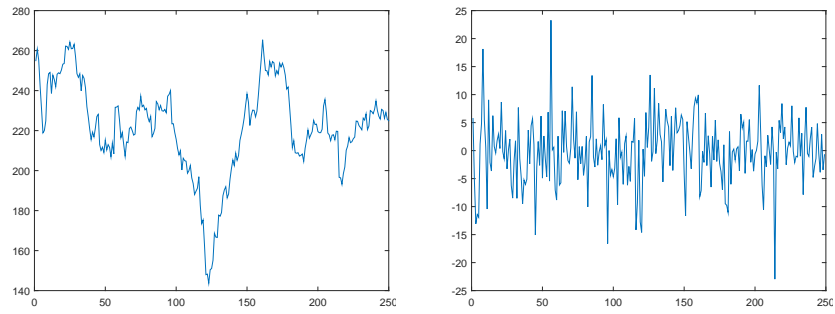
data_toyota = load(filename_toyota);
data_tesla = load(filename_tesla);
%% DATA PLOTS

to = diff(data_toyota); %toyota pnl
te = diff(data_tesla); %tesla pnl

%figure
%scatterhist(diff_toyota, diff_tesla);
%hold off

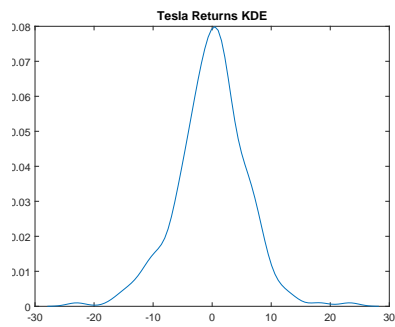
[f,xi] = ksdensity(to);
figure
plot(xi,f);
title('Toyota Returns KDE')
hold off

[f2,xi2] = ksdensity(te);
figure
plot(xi2,f2);
title('Tesla Returns KDE')
hold off
}
```



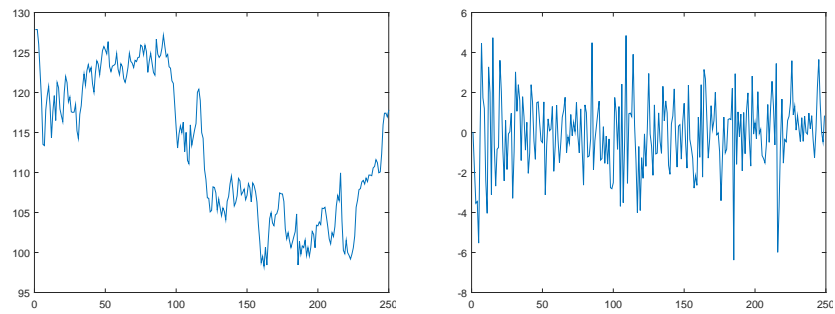
(a) Tesla Stock Price

(b) Tesla P&L



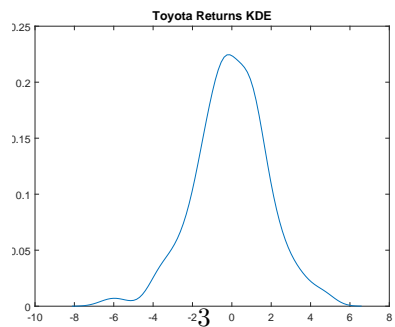
(c) Tesla P&L Distribution

Figure 1: Tesla Data Visualization Aug 2015 - Aug 2016



(a) Toyota Stock Price

(b) Toyota P&L



(c) Toyota P&L Distribution

Figure 2: Toyota Data Visualization Aug 2015 - Aug 2016

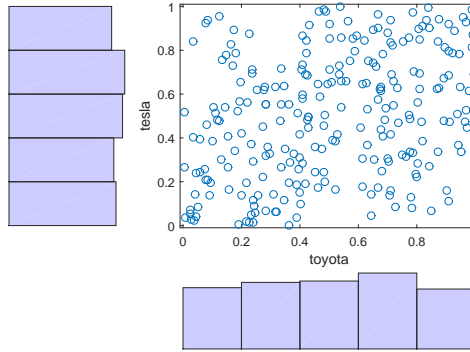


Figure 3: Copula Scatterplot Unit Scaled

3 Copula Fitting

Next, I transform the data to copula scale (within a unit square) using the kernel estimator of the CDF (Fig 3). From these parameters u, v I can fit a t -copula by estimating its parameters ρ and ν . Using Matlab's built in copulafit function which utilizes a Maximum Likelihood Estimation, we observe $\rho = 0.3395$ and $\nu = 3.8699e + 06$.

Because of our relatively small sample size ($n = 249$), the MLE technique chooses too many degrees of freedom. In the case that $\nu = 3.8699e + 06$, the t -copula would simply act as a Gaussian copula. Therefore, in order to continue this study I arbitrarily choose 6 degrees of freedom for the purpose of further study on t -copula.

```
%% FIT T-COPULA TO PNL DATA

u = ksdensity(to,to,'function','cdf');
v = ksdensity(te,te,'function','cdf');

[rho,nu] = copulafit('t',[u,v],'Method','ApproximateML');
```

4 Simulation

Next, we take our t -copula and generate random samples from our new joint distribution function. I chose $N = 10,000$ as our sample size for simulation.

```
%% GENERATE RANDOM SAMPLE FROM T-COPULA
dof = 6;
N = 10000;
r = copularnd('t',rho,dof,N);
u1 = r(:,1);
```

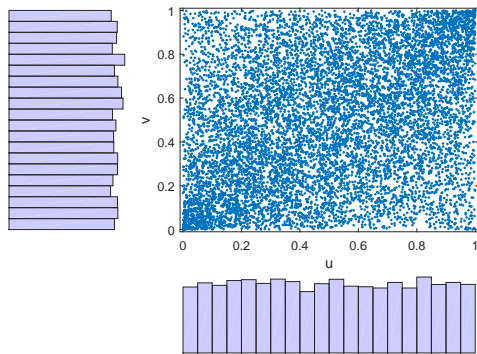


Figure 4: Simulation Results Drawn from Copula

```
v1 = r(:,2);

figure
scatterhist(u1,v1)
xlabel('u')
ylabel('v')
set(get(gca,'children'),'marker','.')

```

We then transform the random sample back to the original scale of the data. See Fig 5.

```
% TRANSFORM RANDOM SAMPLE BACK TO ORIGINAL SCALE OF DATA
x1 = ksdensity(to,u1,'function','icdf');
y1 = ksdensity(te,v1,'function','icdf');

figure
scatterhist(x1,y1)
set(get(gca,'children'),'marker','.')
hold off

figure
combined = [x1; y1];
plot(ksdensity(combined));
hold off

```

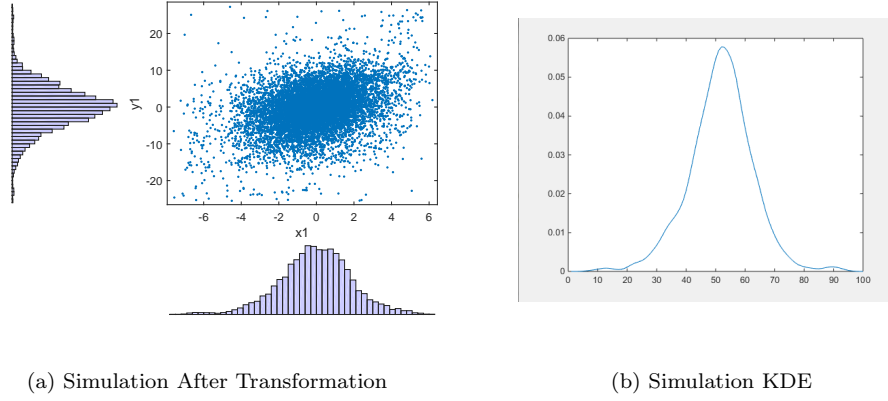


Figure 5: Post Transformation Results

5 Results

From our given simulation, I calculate the 99.97% VaR and Expected Shortfall at the 2% Tail, where ES is defined as

$$ES = \frac{1}{T} \sum_{i=1}^T P_i \quad (4)$$

where i is the ranking of the combined P&L, with $i = 1$ corresponding to the worst loss simulated, and where T is the tail size ($N * 2\%$). Results show 99.97% VaR = -30.4138, and Expected Shortfall = 23.4313.

```
%% CALCULATE VAR at 99.97%
sorted = sort(combined);
var_cutoff_index = int32(.0003*N);
var_combined = sorted(var_cutoff_index);

%% CALCULATE EXPECTED SHORTFALL (2% TAIL)
T = 0.02 * N;
sum=0;
for i = 1:T
    sum = sum + sorted(i);
end
ES_base = sum/T;
```

6 Sensitivity Testing

In this section I conduct the same sensitivity test as the t -copula test in the Ongoing Monitoring Plan of the Trading Risk Economic Capital Model. Our model is calibrated with a parameter choice of $\rho = 0.3395$ and $\nu = 6$. Sensitivity testing of these parameters will be conducted by applying varying parameters, with $\rho = 0\%, 40\%, 99\%$ and $\nu = 4, 6, 10$ and seeing the impact on Expected Shortfall. Refer to Table 1 for results.

ρ	ν	ES	Percent Dif from Base ES
0%	4	-18.2838	+4.97%
0%	6	-18.1657	+5.58%
0%	10	-17.8680	+7.13%
40%	4	-20.2574	-5.29%
40%	6	-20.6726	-7.45%
40%	10	-19.7994	-2.91%
99%	4	-21.8979	-13.82%
99%	6	-22.5536	-17.23%
99%	10	-23.4313	-21.79%

Table 1: Sensitivity Testing

```

%% SENSITIVITY TEST
rho_array = [0 .4 .99];
nu_array = [4 6 10];

ES_array = [];
percent_dif_array = [];

for i = 1:length(rho_array)
for j = 1:length(nu_array)
dof = nu_array(j);
rho = rho_array(i);
N = 10000;
r = copularnd('t',rho,dof,N);
u1 = r(:,1);
v1 = r(:,2);
x1 = ksdensity(to,u1,'function','icdf');
y1 = ksdensity(te,v1,'function','icdf');
combined = x1+y1;
sorted = sort(combined);
T = 0.02 * N;
sum=0;
for k = 1:T
sum = sum + sorted(k);
end
ES = sum/T;
percent_dif = (ES_base - ES) / ES_base;
ES_array = [ES_array ES];
percent_dif_array = [percent_dif_array percent_dif];

end
end

```


7 Questions

Some pending questions that remain unanswered for me:

- Why does MLE estimate such an insanely high DoF and how can I measure the impact of my arbitrary choice of 6 DoF?
 - Ans: Likely because a Gaussian copula is a better choice in this instance. Or the data sample size may be too small.
- Would Gaussian/ other distribution copula shapes have performed similarly or better and how can I measure this?
 - Ans: With financial data, Gaussian or t -copula choices are likely the most appropriate choices.
- How do I determine an appropriate sample size quantitatively?
 - Ans: Usually subjective. More data for sample size is better. For simulation, make sure results converge. Use a confidence interval band to measure effects quantitatively.
- Should I have logged the original P&L data to obtain a more Gaussian distribution before starting to estimate the copula?
 - Ans: In this instance, a log was not needed because the time horizon was short and the P&L was therefore stationary. Data must be stationary before building a copula.

References

- [1] Attilio Meucci *A Short, Comprehensive, Practice Guide to Copulas*, Oct 2011, http://www.garp.org/media/691726/quant_classroom_oct2011.pdf
- [2] David X. Li, *On Default Correlation: A Copula Function Approach*, 2000, <https://cyrusfarivar.com/docs/li.defaultcorrelation.pdf>
- [3] Svetlozar T. Rachev, *Copula Concepts in Financial Markets*, http://statistik.econ.kit.edu/download/Copula_Concepts.in.Financial_Markets.pdf
- [4] Arthur Charpentier, *The Estimation of Copulas: Theory and Practice*, <http://www.crest.fr/ckfinder/userfiles/files/pageperso/fermania/chapter-book-copula-density-estimation.pdf>
- [5] Eric Bouye, *Copulas for Finance A Reading Guide and Some Applications*, 2000, <http://www.thierry-roncalli.com/download/copula-survey.pdf>
- [6] Mathworks Documentation, *Nonparametric Estimates of Cumulative Distribution Functions and Their Inverses*, <http://www.mathworks.com/help/stats/examples/nonparametric-estimates-of-cumulative-distribution-functions-and-their-inverses.html>
- [7] Mathworks Documentation, *Simulating Dependent Random Variables Using Copulas*, <http://www.mathworks.com/help/stats/examples/simulating-dependent-random-variables-using-copulas.html>