UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
FACULDADE DE CIÊNCIAS ECONÔMICAS
DEPARTAMENTO DE ECONOMIA E RELAÇÕES INTERNACIONAIS

GUSTAVO ALOVISI

WORST CASE MIXTURE-COPULA MEAN-CVAR PORTFOLIO OPTIMIZATION: AN IMPLEMENTATION FOR BRAZILIAN INDEXES

Porto Alegre
2019

GUSTAVO ALOVISI

WORST CASE MIXTURE-COPULA MEAN-CVAR PORTFOLIO OPTIMIZATION: AN IMPLEMENTATION FOR BRAZILIAN INDEXES

Work presented in partial fulfillment of the requirements for the degree of Bachelor in Economics.

Advisor: Prof. Dr. Fernando Augusto Boeira Sabino da Silva

Porto Alegre

2019

CIP - Catalogação na Publicação

Alovisi, Gustavo
Worst Case Mixture-Copula Mean-CVaR Portfolio Optimization: An Implementation for Brazilian Indexes / Gustavo Alovisi. -- 2019.
51 f.
Orientador: Fernando Augusto Boeira Sabino da Silva.

Trabalho de conclusão de curso (Graduação) -- Universidade Federal do Rio Grande do Sul, Faculdade de Ciências Econômicas, Curso de Ciências Econômicas, Porto Alegre, BR-RS, 2019.

1. Portfolio Optimization. 2. Copula. 3. Financial Risk. 4. Econometrics. 5. CVaR. I. Boeira Sabino da Silva, Fernando Augusto, orient. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da UFRGS com os dados fornecidos pelo(a) autor(a).

GUSTAVO ALOVISI

WORST CASE MIXTURE-COPULA MEAN-CVAR PORTFOLIO OPTIMIZATION: AN IMPLEMENTATION FOR BRAZILIAN INDEXES

Work presented in partial fulfillment of the requirements for the degree of Bachelor in Economics.
.

Aprovada em: Porto Alegre, _____de _______de 2019.

BANCA EXAMINADORA:

Prof. Dr. Fernando Augusto Boeira Sabino da Silva – Orientador
UFRGS

Prof. Dr. Nelson Seixas Dos Santos
UFRGS

Prof. Dr. Carlos Eduardo Schonerwald da Silva
UFRGS

"Luck is unreliable."

— Amanda Ripley

ABSTRACT

Using data consisting of Brazilian indexes available from 1993 to 2019 on *Economatica's* platform, we employ a Mean-CVaR portfolio optimization through the use of a mixture of multidimensional Clayton, t and Gumbel copula for modelling dependence between assets and an ARMA-GARCH model for univariate fitting. Given a target return, the methodology focuses on minimizng CVaR as the risk measure in replacement of variance used in traditional Markowitz optimization frameworks. We implement a dynamic investing strategy where portfolios are optimized using a rolling daily calibration window. The out-of-sample performance is evaluated using four different daily target returns for the optimizations and compared against three benchmarks: a Gaussian copula Mean-CVaR, an equally weighted portfolio and IBOV's index. Our empirical analysis shows that the Mixture Copula Mean-CVaR portfolio generates a portfolio with better downside risk statistics and lesser drawdowns, with annualized returns similar or better than returns presented in the benchmarks.

JEL classification: G11, G17, G32.
Keywords: Portfolio Choice. Financial Risk. Optimization. Copula. Computational Finance. Econometrics.

RESUMO

Utilizando dados de índices brasileiros disponíveis na plataforma Economatica de 1993 a 2019, nós realizamos uma otimização de portfólio de Média-CVaR a partir do uso de uma mistura de cópulas multidimensionais Clayton, t e Gumbel para modelagem da dependência entre os ativos e um modelo ARMA-GARCH para ajuste univiariado. Dado um retorno alvo, a metodologia foca em minimizar o CVaR como medida de risco em substituição da variância, utilizada em modelos tradicionais de Markowitz de otimização de carteiras. Nós implementamos uma estratégia dinâmica de investimento na qual os portfólios são otimizados utilizando uma janela móvel diária de calibração. A performance fora-da-amostra é avaliada utilizando quatro retornos-alvo diferentes para as otimizações e comparada com três benchmarks: um portfólio de Média-CVaR com cópula Gaussiana, um portfólio com pesos iguais para cada ativo e o índice IBOV. A análise empírica mostra que a otimização de Média-CVaR com mistura de cópulas gera portfólios com melhor *downside risk* e menores *drawdowns*, com retornos anualizados similares ou melhores que os retornos dos benchmarks.

Classificação JEL: G11, G17, G32.

Palavras-chave: Escolha de Portfolios. Risco Financeiro. Otimização. Copulas. Finanças Computacionais. Econometria.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

VaR: Value-at-Risk

CVaR: Conditional Value-at-Risk

LP: Linear Program

MPT: Modern Portfolio Theory

CONTENTS

1 INTRODUCTION

Two types of decision-making frameworks are typically adopted in financial optimization: the utility maximization and the return-risk trade-off analysis. As trade-off analysis explicitly specifies and quantifies risk as a real number, empirically, it is the most utilized optimization framework (ZHU; FUKUSHIMA, 2009). As an important trade-off financial analysis, Markowitz (1952) in his seminal paper "Portfolio Selection" introduces the Modern Portfolio Theory (MPT), or mean-variance framework. In MPT, the estimation of portfolio's return and risk is given by the mean of the expected returns and variance, respectively. He identified that by diversifying a portfolio among different assets and return patterns, one can build an efficient portfolio, with either (i) maximum expected return for a given level of risk (ii) minimum risk for a given level of portfolio returns.

As pointed out by Black and Litterman (1992), however, in the classical mean-variance analysis, the portfolio weights allocation is very sensitive to the mean and co-variance matrix. The authors showed that a small change in the mean can produce a large change in portfolio's decision. Therefore, risk modelling necessity arises as there is uncertainty of the underlying probability distribution of the assets. Kakouris and Rustem (2014) argue that the most common measure for the estimation of portfolio returns remains the expected (mean) return, but many other ways of calculating risk have been developed, particularly following 1990s decade. As Pfaff (2012) elaborated, through 1990s and 2000s the focus of investors shifted, not only because of the financial crisis therein, but also by the need of adequately measuring risks and potential losses during more tranquil market episodes.

Downside risk, as defined in Sortino and Meer (1991), is the risk of asset's actual return being lower than the expected return as well as the uncertainty of the magnitude of this difference. Variance is not a downside risk measure because it measures the both the upside and the downside portfolio risk. Given the need of different risk measures, Value-at-Risk (VaR), a measure of downside risk, has become very popular in financial risk management over the last decades (MORGAN et al., 1996). However, as explained by Zhu and Fukushima (2009), VaR has been criticized in recent years mainly by three aspects. First, VaR is not subadditive in the general distribution case, which means that the portfolio risk is bigger than the sum of the single risk measures of the assets contained in the portfolio. By violating subadditivity, it is not a coherent risk measure as defined by

Artzner et al. (1999). Second, it is non linear and not a convex measure of risk and thus it may not have a single global extrema. Third, it gives a percentile of loss distribution that does not provide an adequate picture of the possible losses in the entire tail of the distribution.

Conditional Value-at-Risk (CvaR), presented by Artzner et al. (1999) and Szego (2005) can be defined as the mean of the tail distribution exceeding VaR. As a measure of risk, CVaR exhibits better properties than VaR and it is further explored by Rockafellar and Uryasev (2000) and Rockafellar and Uryasev (2002). The authors showed that that minimizing CVaR can be achieved by minimizing an auxiliary and more tractable function which computes VaR and CVaR simultaneously, given that outright numerical optimization of CVaR is difficult due to its dependency on VaR. It is also shown that CVaR is a convex function for continuous and discrete distributions, with optimization problems that can be reduced to linear programming, presenting efficient algorithms to optimize portfolios with large dimensions.

Following the ground work for CVaR optimization provided by the authors, one has to make assumptions on the underlying distribution of the assets composing the portfolio. Zhu and Fukushima (2009) model the distribution issue assuming an uncertainty domain (like a ellipsoidal set) in which all feasible uncertainty values lie and applying a robust worst-case technique. However, they also state that assuming a multivariate distribution between the assets is possible. Gaussian distribution is the most commonly used multivariate case, but the use of Gaussian distribution to represent dependence between assets means not respecting financial markets stylized facts about non-normality of returns data (PFAFF, 2012). Hu (2006) and Kakouris and Rustem (2014) explain that the use of Gaussian distribution implies that the probability of losses is the same as the probability of gains, but in the context of financial markets, assets exhibit stronger comovements during crises. One can also use linear correlation as a dependence measure. However, linear correlation correctly depicts the dependence between the random variable of returns only if these are jointly elliptically distributed (SZEGO, 2005; ARTZNER et al., 1999; PFAFF, 2012).

To address the dependence question, Kakouris and Rustem (2014) propose fitting to data a mixture of archimedean copulas. Copulas are multivariate distribution functions whose one-dimensional margins are uniformly distributed on the closed interval [0,1]. Also, copulas consider the dependency between the marginal distribution of the random variables instead of the dependency between the random variables themselves. An important feature of using copulas is that you can separate the selection of multivariate dependency from the selection of the univariate distributions. (CHERUBINI; LUCIANO; VECCHIATO, 2004; NELSEN, 2000). A mixture of copula functions allows modelling a wider range of possible multivariate dependence structure of the assets. In this work, it's applied the Worst-Case Copula CVaR optimization described by Kakouris and Rustem (2014) to Brazilian indexes from January 3, 1993 to June 27, 2019.

The paper is organized as follows: Section 2 presents a mathematical revision of the different CVaR optimization frameworks as well as the final model specification; Section 3 will display the data and empirical methodology used for the implementation of the model; Section 4 provides results and comparison between optimizations; Section 5 concludes the study.

2 MODEL ESPECIFICATION

2.1 Conditional Value-at-Risk

Pflug (2000) defines CVaR and it's optimization problem as follows. Let Y be a stochastic vector standing for market uncertainties and F_Y be its distribution function, i.e., $F_Y(u) = P(Y \leq u)$. Let also $F_Y^{-1}(v) = inf\{u : F_Y(u) \geq v\}$ be its right continuous inverse and assume that it has a probability density function represented by $p(r)$. Define VaR_β as the β-quantile by

$$\begin{aligned} VaR_\beta(Y) &= \text{argmin}\{\alpha \in \mathbb{R} : P(Y \leq \alpha) \geq \beta\} \\ &= F_Y^{-1}(\beta), \end{aligned} \tag{2.1}$$

and the $CVaR_\beta$ as the solution to the following optimization problem:

$$CVaR_\beta = inf\{\alpha \in \mathbb{R} : \alpha + \frac{1}{1-\beta}\mathbb{E}[Y - \alpha]^+\}. \tag{2.2}$$

An easier interpretation of CVaR is given byRockafellar and Uryasev (2000) and Artzner et al. (1999). They have shown that the $CVaR$ is the conditional expectation of Y given that $Y \geq VaR_\beta$, that is,

$$CVaR_\beta(Y) = \mathbb{E}(Y|Y \geq VaR_\beta(Y)). \tag{2.3}$$

In a numerical approach, CVaR function at a confidence level β can be written as

$$CVaR_\beta = \frac{1}{1-\beta}\int_{f(w,r)\leq VaR_\beta(w)} f(w,r)p(r)\,dr, \tag{2.4}$$

where $f(w, r)$ is defined as a loss function depending upon a decision vector w that belongs to any arbitrarily chosen subset $X \in \mathbb{R}^m$ and a random vector $r \in \mathbb{R}^m$. In a portfolio optimization problem, as in Silva and Ziegelmann (2017), the decision vector w can be a vector of portfolio's weights, X a set of feasible portfolios subject to linear constraints and r a vector that stands for market variables that can affect the loss of the assets (a vector of random log returns of each asset).

Given equation 2.4, its clear that CVaR optimization uses VaR in its definition. As stated before, VaR is not convex nor linear. Rockafellar and Uryasev (2000) in their main contribution define a simpler auxiliary function which can be used to calculate CVaR without any need to compute VaR first:

$$F_\beta(w, \alpha) = \alpha + \frac{1}{1-\beta} \int_{f(w,r)\geq\alpha} (f(w,r) - \alpha)\, p(r)\, dr, \tag{2.5}$$

where

$$F(w, \alpha) = \int_{f(w,r)\leq\alpha} p(r)\, dr \tag{2.6}$$

is the non-decreasing and right-continuous cumulative distribution function for the loss function $f(w, r)$ with respect to α. It is also shown that $F_\beta(w, \alpha)$ is convex with respect to α.

Artzner et al. (1999) defines a risk measure as coherent if it satisfies four axioms:

- monotonicity
- translation invariance
- positive homogeneity
- sub-additivity.

Pfaff (2012) gives a brief explanation of the axioms. Let ρ denote a risk measure and $\rho(L)$ the risk value of a portfolio, where the loss L is a random variable.

The axiom of monotonicity requires for two given losses, L_1 and L_2, that $L_1 \leq L_2 \implies \rho(L_1) \leq \rho(L_2)$. The axiom of translation invariance ensres that re risk measure is defined in the same unites as the losses and if formally written as $\rho(L_1) = \rho(L) + l,\ l \in \mathbb{R}$. The axiom of positive homogeneity is satisfied if $\rho(\lambda L) = \lambda\rho(L), \lambda > 0$. This axiom would be violated if the size of a portfolio position directly influenced its riskiness. Finally, the axiom of sub-additivity states that $\rho(L_1 + L_2) \leq \rho(L_1) + \rho(L_2)$. Intuitively, it means that the portfolio risk shall be less than or equal to the sum of the single risk measures of the assets contained in portfolio, i.e., due to diversification effects, the holding of a portfolio is less risky.

Acerbi and Tasche (2002) give a formal proof of the coherence of CVaR as a risk measure, i.e, satisfying the axioms above. Also, WCVaR is said to be a coherent risk measure, and it is demonstrated by Zhu and Fukushima (2009). As stated before, VaR is not coherent because sub-additivity does not hold.

2.2 CVaR Optimization Problem

Following Wuertz et al. (2010), an intuitive mean-CVaR optimization problem can be written as

$$\begin{aligned} \min_{w \in X} \quad & CVaR_\beta(w) \\ \text{s.t} \quad & w^T \hat{\mu} = R \\ & w^T 1 = 1, \end{aligned} \tag{2.7}$$

where R is a given target return, w is an assets weights vector an β the desired CVaR's significancy.

Rockafellar and Uryasev (2000) give a formal proof that by minimizing the auxiliary function $F_\beta(w, \alpha)$, one can find the optimal weights of the portfolio, CVaR and VaR simultaneously over an feasible set, i.e.,

$$\min_{w \in X} CVaR_\beta(w) = \min_{(w,\alpha) \in X \times R} F_\beta(w, \alpha). \tag{2.8}$$

They also develop a method to approximate $F_\beta(w, \alpha)$ using J scenarios, $r_j, \quad j = 1, ..., J$, sampled from the density function $p(r)$, assuming that the analytical representation for the density $p(r)$ in (2.5) is not available. Then, it's possible to approximate

$$\begin{aligned} F_\beta(w, \alpha) &= \alpha + \frac{1}{1-\beta} \int_{f(w,r) \geq \alpha} (f(w,r) - \alpha) \, p(r) \, dr \\ &= \alpha + \frac{1}{1-\beta} \int_{r \in R^m} (f(w,r) - \alpha)^+ \, p(r) \, dr, \end{aligned} \tag{2.9}$$

where $z^+ = max(z, 0)$, by its discretized version:

$$F_\beta^d(w, \alpha) = \alpha + \frac{1}{(1-\beta)J} \sum_{j=1}^{J} (f(w, r_j) - \alpha)^+. \tag{2.10}$$

If it's assumed that the feasible set X and the loss function $f(w, r_j)$ are convex, it's possible to solve the following convex optimization problem:

$$\min_{w \in X, \alpha \in \mathbb{R}} F_\beta^d(w, \alpha). \tag{2.11}$$

By solving (2.11) we can then obtain the optimal portfolio vector of weights, w^*, the portfolio's corresponding VaR, α^*, and the optimal CVaR, which equals to $F^d(w^*, \alpha)$. Finally, if the loss function $f(w, r_j)$ is linear with respect to w and the set X is given by linear inequalities, then we can reduce (2.11) to the Linear Problem (ROCKAFELLAR; URYASEV, 2002):

$$\begin{aligned}
\min_{w\in R^n, z\in\mathbb{R}^J, \alpha\in\mathbb{R}} \quad & \alpha + \frac{1}{(1-\beta)J}\sum_{j=1}^{J} z_j \\
\text{s.t} \quad & z_j \geq f(w, r_j) - \alpha, \quad j = 1, ..., J, \\
& z_j \geq 0, \quad j = 1, ..., J, \\
& w \in X, \\
& w^T = 1, \\
& w^T\hat{\mu} = R.
\end{aligned} \tag{2.12}$$

This is done adding auxiliary variables, z_j, to replace $(f(w, r_j) - \alpha)^+$, imposing the constraints $z_j \geq f(w, r_j) - \alpha$ and $z_j \geq 0$.

Therefore, CVaR optimization can be carried out using efficient Linear Programming algorithms (for instance Simplex), provided that $f(w, r_j)$ is linear with respect to w and both the feasible set X and the loss function $f(w, r_j)$ are convex.

2.3 Copulas

The theory of Copula functions was first introduced by Sklar (1959) and by mid-1990s, it started being used as a tool for modelling dependencies between assets in empirical finance (PFAFF, 2012). Copulas are multivariate distribution functions whose one-dimensional margins are uniformly distributed on $[0, 1]$. A copula C is a function such that

$$C(u_1, ..., u_n) = P(U_1 \leq u_i, ..., U_n \leq u_n) \tag{2.13}$$

where $U_i \;\; U[0, 1]$ and u_i are realizations of U_i for $i = 1, ..., n$. It is possible to use copulas to replace probability distribution functions due to Sklar's theorem and its corollary:

Theorem 2.3.1 (Sklar's Theorem) *Let F be an n-dimensional distribution function with margins $F_1, ...F_n$. Then there exists an n-copula C such that for all $x \in R^n$,*

$$F(x_1, ..., x_n) = C(F_1(x_1), ..., F_n(x_n)). \tag{2.14}$$

Furthermore, if $F_1, ...F_n$ are continuous, then C is unique.

Corollary 2.3.1.1 *Let F be an n-dimensional distribution function with margins $F_1, ...F_n$ and let C be and n-copula. Then, for any $u = (u_1, ..., u_n) \in U[0,1]^n$,*

$$C(u_1, ...u_n) = F(F_1^{-1}(u_1), ..., F_n^{-1}(u_n)), \tag{2.15}$$

where $F_i^{-1}, i = 1, ..., n$ are the quasi-inverses of the marginals.

Kakouris and Rustem (2014) show that by using Theorem 2.3.1 and Corollary 2.3.1.1, it is possible to derive a relation between the probability density functions and the copulas. They define the copula density of an n-copula as:

Definition 2.3.1 *Let f be the multivariate probability density function the probability distribution F and $f_1, ..., f_n$ the univariate probability density functions of the margins $F_1, ..., F_n$. The copulas density function of an n-copula C is the function c: $U[0,1]^n \mapsto [0, \infty)$ such that*

$$c(u_1, ..., u_n) = \frac{\partial^n C(u_1, ..., u_n)}{\partial u_1, ..., \partial u_n} = \frac{f(x_1, ..., x_n)}{\prod_{i=1}^n f_i(x_i)}. \tag{2.16}$$

The definition allows us to separate the modelling of the marginals $F_i(x_i)$ from the dependence structure represented by C. The copula probability density function is then the ratio of the join probability function to what it would have been under independence. Following Silva and Ziegelmann (2017) interpretation, it's possible to consider the copula as the adjustment that we need to make in order to convert the independent probability density function into the multivariate density function. I.e., copulas decompose the joint p.d.f from its margins. As in Hofert et al. (2018b), we can estimate the multivariate distribution in two parts: (i) finding the marginal distribution of for each x_i (ii) finding the dependency between the filtered data from (i). This methodology allows us to derive join distributions from the marginal independently from each other, as no assumption of the joint behavior of the marginals is needed. As such, modelling with copula functions provides a great deal of flexibility in regard to joint distribution models. An extensive review of copula modelling in econometrics and finance can be found on Fan and Patton (2014) and Patton (2008).

As presented in Nelsen (2000) and Pfaff (2012), we consider two copula distinct copula families to model joint p.d.fs: Archimedean Copulas and Elliptical Copulas. An Elliptical Copula, like Gaussian/Normal or t-student, is a copula in which the dependence between the random variables are captured implicitly by a distribution parameter. Elliptical copulas are simple and easy to simulate, however, they are symmetric and this can be a problem, since given stylized facts of financial data, empirical distributions of the loss functions are generally skewed. Gaussian copula is one of the most used copula in modelling. Regarding risk modelling, it has a limited application since a characteristic of Gauss copula is having zero tail dependency. A second example of elliptical copula is Student's t copula. In contrast to Gaussian copula, t-copula's tail dependency is not zero, it has a symmetric lower and upper tail dependency. However, the use of a single t-copula to financial modelling can also be problematic, since another financial data stylized fact is that returns gains and losses are not symmetric: there is a stronger comovement of financial assets in crises periods.

An Archimedean copula is defined as

$$C(u_1, ...u_n) = \Psi^{-1}(\Psi(u_1) + ... + \Psi(u_n)), \tag{2.17}$$

where Ψ is a copula-generating function. Extended literatureregaring Archimedian copulas can be found in Cherubini, Luciano and Vecchiato (2004); Hofert et al. (2018b); Nelsen (2006). As opposed to Elliptical copulas, Archimedian copulas are not necessarily symmetric, since tail dependency is modelled via the particular copula-generating function. Two useful and well-known Archimedean copulas are Clayton Copula and Gumbel Copula, each having their characteristics.

A Clayton copula have its copula-generation function given by

$$\Psi(t) = (t^{-\delta} - 1)/\delta, \quad \delta \in (0, \infty), \tag{2.18}$$

where $\delta \longrightarrow \infty$ implies perfect lower tail dependency and $\delta \longrightarrow \infty$ implies lower tail independence.

A Gumbel copula is a copula with generating function

$$\Psi(t) = (-\ln t)^{\theta}, \quad \theta \geq 1, \tag{2.19}$$

where perfect upper tail dependence exists for $\theta \longrightarrow \infty$ and upper tail independence exists for $\theta \longrightarrow 1$.

In this work, a linear combination of the best mixture of Clayton, t and Gumbel copula is considered for each optimization period. This mixture is chosen because these three copulas have different tail dependence characteristics. It combines a copula with lower tail dependence, a symmetric tail dependence and an upper tail dependence. Hence, a wide range of possible dependence structures are considered within the model, with the aim to capture best the dependence between the individual assets contained in the portfolio. The mixture is derived by the mixture used in the works of Pfaff (2012), Kakouris and Rustem (2014) and Hu (2006), but using an Elliptical t-copula instead of an Archimedean Frankel copula.

2.4 Worst Case Copula CVaR

Now that the CVaR optimization problem for a random vector of distributions have been defined, as well as theorems to associate copulas and these distributions, it is possible to define a Worst Case CVaR optimization with (mixture) copula functions. This is accomplished following Kakouris and Rustem (2014), which gives us the Worst Case CVaR using copula functions, and Zhu and Fukushima (2009), which gives us Worst Case CVaR using a mixture of distributions.

Let $w \in W$ be a decision vector, $u \in U[0,1]^n$ a random vector that follows a continuous distribution with copula density function $c(.)$ and $F(r) = (F_1(r_1), ..., F_n(x_n))$ a set of marginal distributions where u = F(r). The copula corresponding equation of 2.9 is

$$\begin{aligned} G_B(w,\alpha) &= \alpha + \frac{1}{1-\beta} \int_{f(w,u)\geq\alpha} (f(w,u)-\alpha)\, c(u)\, du \\ &= \alpha + \frac{1}{1-\beta} \int_{u\in U[0,1]^n} (f(w,u)-\alpha)^+\, c(u)\, du, \end{aligned} \tag{2.20}$$

where the discretized version following K scenarios is represented as

$$G^d_\beta(w,\alpha) = \alpha + \frac{1}{(1-\beta)K} \sum_{j=1}^{K} (f(w,u_j)-\alpha)^+. \tag{2.21}$$

Using a mixture of copulas,

$$C(.) \in C_{mix} = \{\sum_{i=1}^{l} \pi_i C_i(.) : \sum_{i=1}^{d} \pi_i = 1, \pi_i \geq 0, i = 1, ...l\}, \tag{2.22}$$

we can then express equation (2.20) as an equation for for evaluation using Monte Carlo simulations. We do this by sampling realizations from the copulas $C_i(.)$ using as inputs the filtered uniform margins,

$$G_\beta^d(w,\alpha) = \alpha + \frac{1}{(1-\beta)K^i}\sum_{j=1}^{K}(f(w,u_j^i)-\alpha)^+,\ i=1,2...l, \tag{2.23}$$

where u_j^i and K^i are the j-th sample drawn from the copula $C_i(.)$ of the mixture copula using as inputs the filtered uniform margins and its corresponding size, respectively.

Following assumptions of convexity and linearity of the loss function $f(w,u)$ with respect to w, the optimization problem,

$$\min_{w\in W,\alpha\in\mathbb{R}} G_\beta^d(w,\alpha), \tag{2.24}$$

can be modelled following Rockafellar and Uryasev (2002) mean-CVaR approach as

$$\begin{aligned} \min_{x\in\mathbb{R}^n,z\in\mathbb{R}^K,\alpha\in\mathbb{R}} \ & \alpha + \frac{1}{(1-\beta)K^i}\sum_{k=1}^{K^i} z_i, \\ \text{s.t}\quad & z_i \geq f(w,u_k^i)-\alpha, k=1,...,K, \\ & z_i \geq 0, i=1,...,d, \\ & w \in W, \\ & w^T\hat{\mu} \geq R, \\ & w^T 1 = 1. \end{aligned} \tag{2.25}$$

3 METHODOLY AND DATA

The empirical study of the Worst Case Copula-CVaR portfolio optimization outlined above considers a sample data of 8 Brazilian indexes contained in *Economatica's* Platform from January 3, 1993 to June 27, 2019 - a total of 5811 trading days. $L = 5810$ daily log-returns are calculated from the following indexes:

- DOLX19: Commercial Dollar contract,
- DOLOF: Future Dollar contract,
- INDV19: Future IBOV contract,
- IBOV: Ibovespa index,
- IBXX: index of Ibov's 100 most liquid assets,
- IEEX: index of Ibov's most relevant energy companies,
- OZ1D: Gold contract,
- TBF: Brazillian Basic Financial Rate.

For the model estimation, we consider a period $T = 1260$ days. A rolling window optimization approach similarly from Xi (2014) is applied for $L - T = 4550$ optimizations:

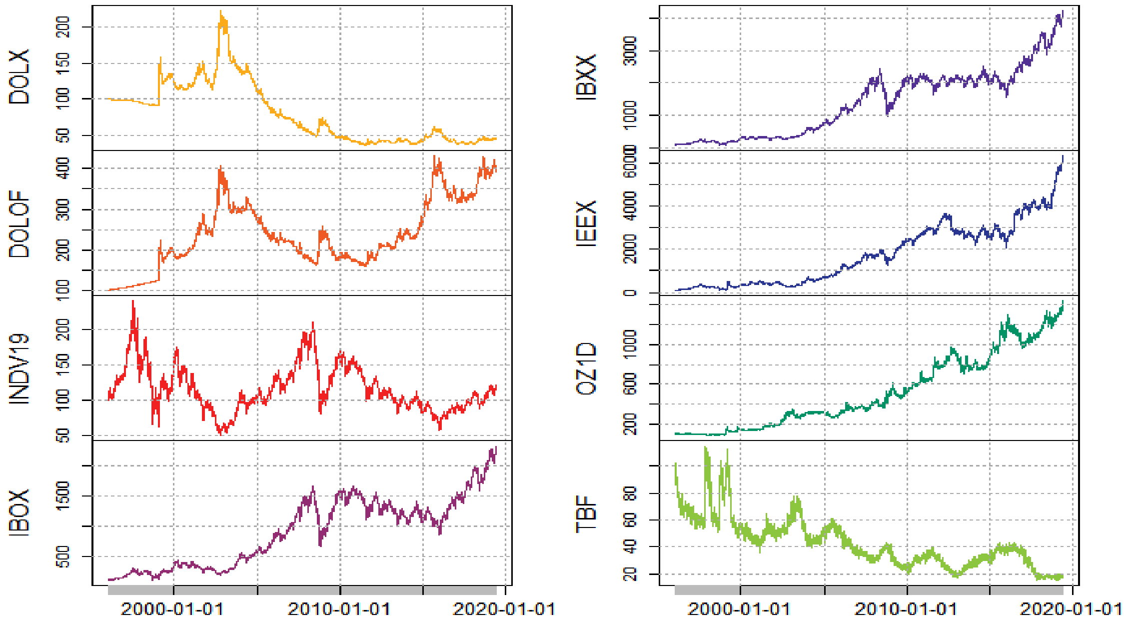
- Optimization 1: Use day 1 to day 1260 to estimate the Worst-Case Copula-CVaR model and determine portfolio weights for day 1261.
- Optimization 2: Use day 2 to day 1261 to estimate the Worst-Case Copula-CVaR model and determine portfolio weights for day 1262.
- ...
- Optimization 4550: Use day 1260 to day 5809 to estimate the Worst-Case Copula-CVaR model and determine portfolio weights for day 5810.

Next, descriptive statistics about the sample are presented.

3.1 Data and descriptive statistics

Figure 3.1 shows the indexes price evolution of the data set for the whole sample period L, where $t = 1$ has base value of 100 for every asset. Figure 3.2 shows the log-return plots of the data. It is worth commenting that DOLX19 and DOLOF present very low returns in the initial period of data. This is due to Brazil's fixed exchange rate regime that lasted until January, 1999. As shown on Table 3.1, all but one of the asset's log

Figure 3.1: Price Indexes



returns distribution have positive skewness, which indicates non-symmetric right skew of returns and attends stylized facts of financial data presented in Pfaff (2012). Regarding kurtosis, all of the assets log returns have positive values. This indicates the log returns distributions have heavier tails than Normal distribution. In particular, IEEX Index has a kurtosis of approximately 111.71, which indicates tails a lot heavier than Normal ones.

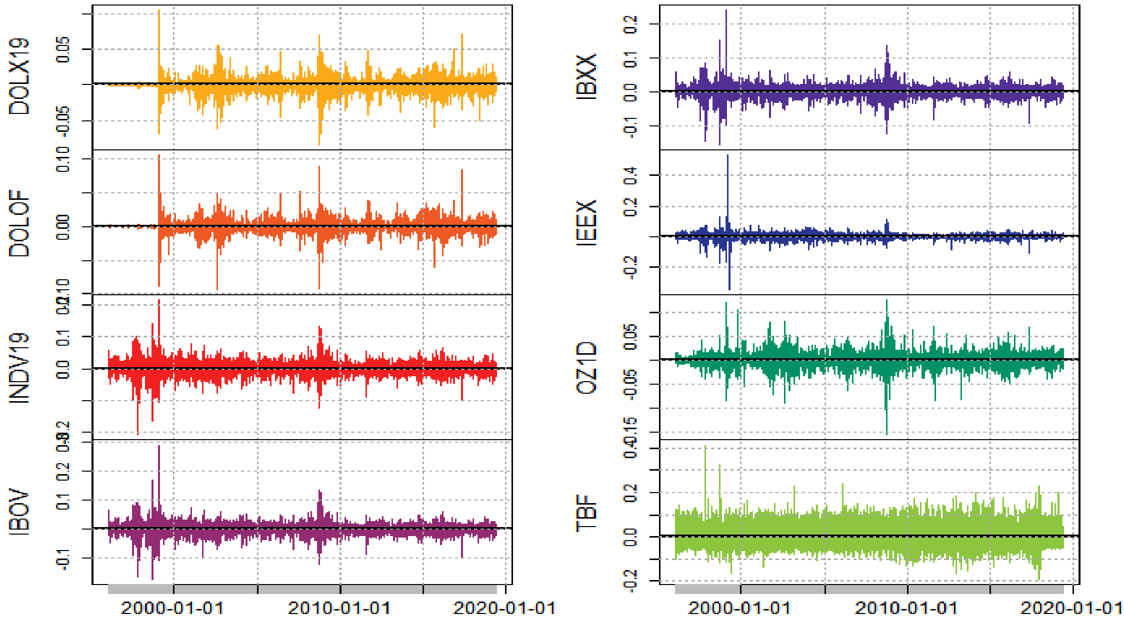
Table 3.1: Descriptive Statistics of log returns

Index	Min	Max	Aritm. Mean	Std.dev	Skewness	Kurtosis
DOLX19	-0.08327	0.10436	-0.00014	0.01023	0.39413	8.87790
DOLOF	-0.09359	0.10529	0.00024	0.00945	0.53673	16.77306
IND19	-0.20634	0.21396	0.00003	0.02099	-0.05957	7.59111
IBOV	-0.17229	0.28818	0.00054	0.01993	0.27245	13.44936
IBXX	-0.15616	0.24109	0.00064	0.01796	0.08856	12.30610
IEEX	-0.34612	0.53201	0.00071	0.01984	2.33441	111.71610
OZ1D	-0.15441	0.12402	0.00046	0.01483	0.20076	8.27108
TBF	-0.19368	0.41310	-0.00029	0.05368	0.76468	1.16543

3.2 Empirical Strategy

To apply the Worst Case Copula-CVaR portfolio optimization, we follow mainly the steps presented in Pfaff (2012), Hofert et al. (2018a), Hofert et al. (2018b) and Xi

Figure 3.2: Log Returns



(2014). These steps are presented below and repeated for every optimization step we previously shown. In addition, we consider for benchmarking an Equal Weight Portfolio (EWP) and a Gaussian Copula Portfolio (GCP).

(1) First, we fit an ARMA(p,q)-GARCH(1,1) model with skewed t-distributed innovations to each univariate time series. Note that $p, q \in [[0, 2]]$, where $[[0, 2]]$ is a discrete interval. The choice of (p, q) is given by the (AR, MA) order that minimizes BIC information criteria for the univariate ARMA model fit of each asset.

(2) Using the estimated parametric model, for each asset we construct standardized residuals vector,

$$\frac{\hat{\epsilon}_{t,j}}{\hat{\sigma}_{t,j}}, \quad t = 1, ..., (L - T) \quad \text{and} \quad j = 1, ..., 8. \tag{3.1}$$

(3) Calculate pseudo-uniform variables from the standardized residuals parametrically using the Skewed-t distribution of the GARCH error process. This can also be done semiparametrically using the empirical distribution functions of the standardized residual vectors, see Pfaff (2012) or Hofert et al. (2018a) for an example.

(4) Estimate the multivariate Clayton-t-Gumbel Mixture Copula model to data that

has been transformed to [0,1] margins from the linear combination of copulas,

$$C^{CtG}(\Theta, u) = \pi_1 C^C(\theta_1, u) + \pi_2 C^t(\theta_2, \theta_3, u) + \pi_3 C^G(\theta_4, u), \tag{3.2}$$

where Θ is the Clayton, t and Gumbel copula parameters, π_i is a copula weight parameter such that $\pi_i \in [0, 1]$ and $\sum \pi_i = 1$ and u is the vector of pseudo-uniform observations for each asset.

The estimation of the copula parameters and weights are jointly obtained by the minimization of the negative log-likelihood of the weighted densities from the Clayton, t and Gumbel copulas. Copula densities are computed as in Hofert et al. (2018b). A general non-linear augmented Lagrange multiplier method solver is employed based on Ye (1987) work.

(5) Use the dependence structure determined by the estimated Copula Mixture for generating $K = 10000$ scenarios of random variates for the pseudo-uniformly distributed variables.

(6) Compute Skewed-t quantiles for these Monte Carlo draws, $z_{j,t}$, for $j = 1, .., 8$ and $t = 1, ..., (L - T)$.

(7) For each j asset, determine the K scenarios of simulated daily log-returns for the out-of-sample following day we are forecasting,

$$r_{j,t} = X_j + \epsilon_{j,t}, \tag{3.3}$$

where X_j is provided by the ARMA(p,q) model,

$$X_{j,t} = \epsilon_{j,t} + \sum_{i=0}^{p} \phi_{j,i} \, X_{j,t-i} + \sum_{i=0}^{q} \theta_{j,i} \, \epsilon_{j,t-1} \tag{3.4}$$

and $\epsilon_{j,i}$ is the error term following a GARCH(1,1) process given as

$$\begin{aligned} \epsilon_{j,t} &= \sigma_{j,t} \, z_{j,t} \\ \sigma^2_{j,t} &= \alpha_{j,0} + \alpha_{j,1} \, \epsilon^2_{j,t-1} + \beta_{j,1} \, \sigma^2_{j,t-1} \end{aligned} \tag{3.5}$$

(8) Finally, use the simulated data as inputs when optimizing portfolio weights by minimizing CVaR for a confidence level of 5% and a given portfolio target return. This is done using the works of Wuertz et al. (2010), in which the Rockafellar and Uryasev (2002) method is applied for optimizing CVaR with a linear program. To access optimization

performance, we run the optimization for target daily returns of 0, 0.012%, 0.024% and 0.036%. This is approximately 0%, 3.07%, 6.23% and 9.5% annually.

Similar steps are applied for optimizing the Gaussian Copula Portfolio for each period. However, as there is no need to estimate a mixture of copulae functions, step (4) of the optimization only fits a Gaussian Multivariate Copula to given pseudo-uniform data following Hofert et al. (2018b) method.

For each period in $(L - T)$, the respective estimated vector of the assets weights and the simple returns of the data-set are used to calculate out-of-sample portfolio returns for MCP and GCP, as

$$R_t^{port} = \sum_{j=1}^{8} w_{j,t} \, r_{j,t}, \quad t = 1, ..., (L - T). \tag{3.6}$$

For EWP, the weight of each asset is simply $1/N$.

Using calculated out-of-sample portfolio returns for Mixture Copula Portfolio, Gaussian Copula Portfolio and Equal Weighted Portfolio, several performance measures are calculated as in Peterson and Carl (2019).

3.3 Performance Measures

The following performance measures are computed in Peterson and Carl (2019) and based on Bacon (2008): annualized mean excess return, annualized standard deviation, $VaR_{0.95}$, $CVaR_{0.95}$, Semi-Deviation, $CDaR_{0.95}$, Worst-Drawdown, Average Drawdown, Average Drawdown Lenght, Annualized Sharpe Ratio, Burke Ratio, Sortino Ratio, Upside Potential, Downside Frequency, Calmar Ratio, Drawdown Deviation and Omega Sharpe Ratio.

Annualized returns are calculated as

$$prod(1 + R)^{scale/n} - 1 = \sqrt{prod(1 + R)^{scale}}^{n} - 1 \tag{3.7}$$

and Annualized Standard Deviation is calculated as

$$An.StdDev = \sqrt{\sigma}\sqrt{252}. \tag{3.8}$$

From an absolute return investor's perspective wishing to avoid losses, Drawdown is a famous measure of risk. Average Drawdown is the average continuous negative return over an investment period, and calculated as

$$Av.DD = |\sum_{j=1}^{j=d} \frac{D_j}{d}|, \tag{3.9}$$

where D_j is the jth drawdown over the entire period and d is the total number of drawdowns in entire period.

The Maximum Drawdown ($WorstDD.$), is the maximum potential loss over the out-of-sample period. It represents the maximum loss an investor can suffer buying at the highest and selling at the portfolio's lowest.

Average Length ($Av.Length$) is similar to Average Drawdown, but calculates length instead of the depth.

Drawdown Deviation ($DDDev.$) calculates a stantard deviation-type statistic using individual drawdowns:

$$DDDev. = \sqrt{\sum_{j=1}^{j=d} \frac{D_j^2}{n}}. \tag{3.10}$$

It is also possible to define a Sharpe-type measure using the maximum drawdown rather than the standard deviation to reflect investor's risk. The $CalmarRatio$ is calculated as

$$CalmarRatio = \frac{An.Return}{WorstDD.}. \tag{3.11}$$

Standard deviation and symmetrical normal distribution are foundations of MPT. However, Standard deviation measures both upside and downside risk, thus, it makes sense to measure the deviation strictly regarding the losses of the portfolio.

Downside Risk measures the variability of underperformance below a minimum accepted return (MAR). It is defined as

$$\sigma_D = \sqrt{\sum_{t=1}^{n} \frac{min[R_i - MAR, 0]^2}{n}}. \tag{3.12}$$

.

Semi-Deviation ($Semi.Dev$) is a special case of σ_D where $MAR = mean(R_i)$.

To calculate Downside Frequency ($DownsideFreq.$) we take the subset of returns that are less than the target (or Minimum Acceptable Returns (MAR)) returns and divide the length of this subset by the total number of returns.

The Omega-Sharpe ratio is a conversion of the omega ratio to a ranking statistic in familiar form to the Sharpe ratio. It is calculated as

$$OmegaSharpe = \frac{R_p - MAR}{\frac{1}{n}\sum_{i=1}^{i=n} max(R_P - MAR, 0)}, \tag{3.13}$$

where R_P is the return of the portfolio and n is the number of observations.

A natural extension of the Omega-Sharpe ratio is Sortino Ratio. The difference is that Sortino uses the second-order Lower Partial Moment instead of the first-order Lower Partial Moment as OmegaSharpe,

$$SortinoRatio = \frac{R_P - MAR}{\sigma_D}. \tag{3.14}$$

Lastly, we also consider tail downside risk measures: $VaR_{0.95}$, $CVaR_{0.95}$ and $CDaR_{0.95}$. VaR and $CVaR$ calculation is already explained. A thorough definition of $CDaR$ can be consulted in Chekhlov, Uryasev and Zabarankin (2004).

4 EMPIRICAL RESULTS

Tables 4.1 and 4.2 computes out-of-sample annualized mean excess return, annualized standard deviation, $VaR_{0.95}$, $CVaR_{0.95}$, Semi-Deviation, $CDaR_{0.95}$, Worst-Drawdown, Average Drawdown, Average Drawdown Lenght, Annualized Sharpe Ratio, Burke Ratio, Sortino Ratio, Upside Potential, Downside Frequency, Calmar Ratio, Drawdown Deviation and Omega Sharpe Ratio for optimizations using daily target returns of 0%, 0.012%, 0.024% and 0.036% respectively. For calculation of risk measures it is considered a Risk Free Rate of 0 and a Minimum Accepted Return of 0.

By analyzing Table 4.1 and Table 4.2, we find that the Worst Case Mixture Copula CVaR optimization offers better hedges against losses than the $1/N$ Portfolio, IBOV Portfolio and Gaussian Copula CVaR Portfolio in every daily target return constrained optimization realized. The Mixture Copula Mean-CVaR portfolio shows in almost every performance measure better downside risk performance, better drawdown performance and better VaR and CVaR while offering similar or greater annual returns than $1/N$, IBOV and Gaussian portfolios. It has also bigger Sortino and Omega Sharpe ratio than the other benchmark portfolios in every target return considered.

When comparing solely the Mixture Copula and the Gaussian Copula Portfolio, it is clear that the former has better performance measures on the different target returns considered. Furthermore, Mixture Portfolio offers consistent performance over the different target returns constraints. An increase in target return is followed by an increase in both Annualized Returns and risk measures, while Gaussian Portfolio seems to be very sensitive regarding the constraint. For 0 and 0.00012 target return, Gaussian Portfolio shows somewhat similar performance measures considering the Mixture one, but as the target return increases, the difference between the two quickly increases as well.

Lastly, Figure 4.1 depicts out-of-sample drawdowns of the considered portfolios for the different target returns of the period. Although the minimization of drawdowns or $CDAR_{0.95}$ is not the aim of the optimization, it's analysis still provides insights about the considered portfolio performance.

Considering the Mixture Portfolio (red line), it is possible to see a relatively big drawdown period in the beginning of the out-of-sample performance, which tends to decrease as the target return increases. This could be due to the need of diversifying into higher return assets in order to fulfill the target return constraint, which consequently leads to a lower drawdown. It is also worth noting that in sensitive moments for global markets and Brazilian financial market (9/11 terrorists attack, 2007-2008 financial crisis, 2016 Brazilian impeachment), Mixture portfolio shows lower drawdowns than other strategies.

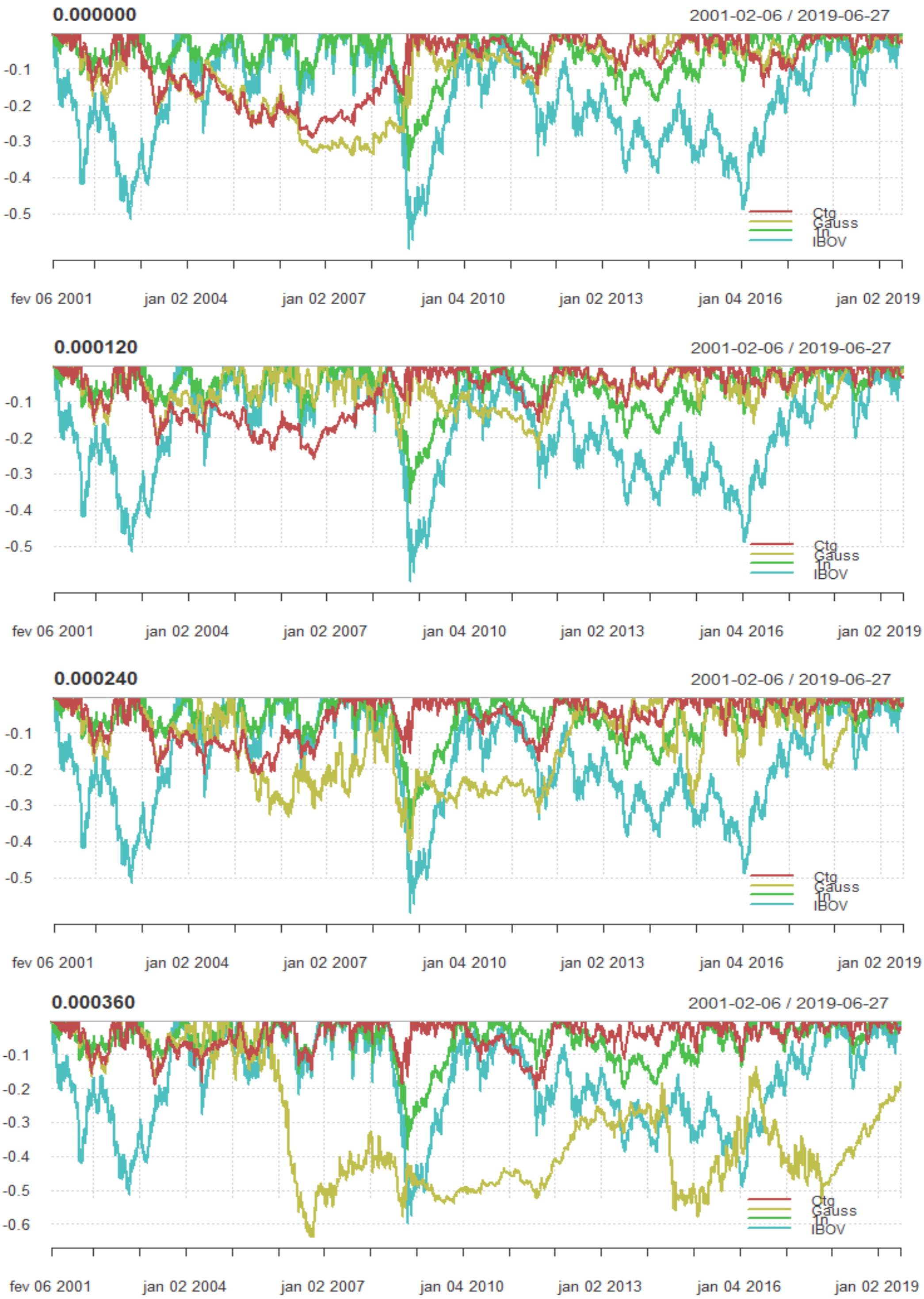
Table 4.1: 1/N Portfolio and IBOV Performance Measures

	1/N	IBOV
An. Return	0.0988	0.1045
An. StdDev	0.1560	0.2660
VaR	0.0152	0.0276
CVaR	0.0215	0.0389
Semi-Dev	0.0069	0.0124
CDaR	0.0712	0.0989
Worst DD	0.3818	0.5996
Av. DD	0.0266	0.0407
Av. Length	26.389	40.844
An. Sharpe	0.6333	0.3774
Burke Ratio	0.1551	0.0871
Sortino	0.0629	0.0451
Upside Potential	0.7745	0.7302
Downside Freq.	0.4798	0.4787
Calmar Ratio	0.2587	0.1743
DD Dev.	0.0099	0.0148
Omega Sharpe	0.1212	0.0893

Table 4.2: Daily Rebalancing Mixture copula portfolio and Gaussian copula portfolio Performance Measures

	.000% Mix	.000% Gauss	.012% Mix	.012% Gauss	.024% Mix	.024% Gauss	.036% Mix	.036% Gauss
An. Return	0.1049	0.0748	0.1125	0.1170	0.1212	0.1265	0.1551	0.0319
An. StdDev	0.1181	0.1237	0.1218	0.1535	0.1286	0.2240	0.1398	0.2724
VaR	0.0107	0.0110	0.0112	0.0135	0.0117	0.0184	0.0133	0.0240
CVaR	0.0167	0.0177	0.0172	0.0219	0.0182	0.0304	0.0194	0.0398
Semi-Dev	0.0052	0.0054	0.0053	0.0066	0.0045	0.0092	0.0061	0.0113
CDaR	0.0512	0.0570	0.0617	0.0580	0.0495	0.1066	0.0532	0.1645
Worst DD	0.2918	0.3403	0.2590	0.2357	0.2160	0.4312	0.2035	0.6388
Av. DD	0.0215	0.0215	0.0228	0.0241	0.0203	0.0417	0.0215	0.0526
Av. Length	29.735	37.084	26.000	25.650	22.251	41.607	19.062	88.569
An. Sharpe	0.8880	0.6044	0.9236	0.7623	0.9424	0.5646	1.1094	0.1173
Burke Ratio	0.2110	0.1464	0.2266	0.1927	0.2307	0.1557	0.2762	0.0312
Sortino	0.0853	0.0603	0.0887	0.0756	0.0904	0.0641	0.1057	0.0241
Upside Potential	0.7200	0.7028	0.7272	0.6724	0.7299	0.6663	0.7548	0.6296
Downside Freq.	0.4655	0.4666	0.4691	0.4641	0.4670	0.4747	0.4664	0.4846
Calmar Ratio	0.3594	0.2198	0.4344	0.4965	0.5611	0.2933	0.7621	0.0501
DD Dev.	0.0075	0.0077	0.0078	0.0092	0.0076	0.0126	0.0087	0.0118
Omega Sharpe	0.1794	0.1285	0.1867	0.1673	0.1887	0.1454	0.2184	0.0546

Figure 4.1: Drawdown Comparison



5 CONCLUDING REMARKS

This work incorporates copula-based dependence structure and time-varying mean and volatility in asset returns using a Copula-ARMA-GARCH model in order to perform a Mean-CVaR portfolio optimization. Using Brazilian indexes data from 1993 to 2019, we evaluate performance of the described portfolio and compare it to three benchmarks: a Gaussian Copula-Mean-CVaR portfolio, an equally weighted ($1/N$) portfolio and IBOV index.

By optimizing Mean-CVaR portfolio with the dependence structure of the assets modelled with a mixture of Clayton, t and Gumbel copula, we found that the resulting portfolio offers better downside-risk measures and drawdown measures than the Gaussian, 1/N and IBOV index portfolio considering the different target returns of the Mean-CVaR optimizations proposed. Moreover, the portfolio in question offers consistent downside-risk measures as the target return constraint increases, as well as similar or bigger annualized returns considering the different portfolio benchmarks presented.

However, we have to bear in mind that the CVaR optimization has been carried out in the absence of transaction costs and using a daily rebalancing strategy. As a consequence, the benefit of the Mixture Copula portfolio over the others can be hindered if high transaction costs are present. As solution to such problems, we may (i) incorporate transaction cost as an additional constraint in the CVaR optimization (ii) reduce rebalancing frequency (iii) compute Turnover measure for the portfolios, where higher Turnover means higher transaction costs.

Further studies could investigate the role of transaction costs and the performance of different rebalancing periods. The employ of canonical vine copulas as in Xi (2014) or time-varying copulas as in Ausin and Lopes (2010) to model dependence can also be investigated. Finally, Reality Checks and Data Snooping tests as Hansen (2005) should be employed to test for predictive ability.

REFERENCES

ACERBI, C.; TASCHE, D. On the coherence of expected shortfall. **Journal of Banking Finance**, v. 26, n. 7, p. 1487–1503, 2002. Available from Internet: <https://EconPapers.repec.org/RePEc:eee:jbfina:v:26:y:2002:i:7:p:1487-1503>.

ARTZNER, P. et al. Coherent measures of risk. **Mathematical Finance**, v. 9, n. 3, p. 203–228, 1999. Available from Internet: <https://EconPapers.repec.org/RePEc:bla:mathfi:v:9:y:1999:i:3:p:203-228>.

AUSIN, M. C.; LOPES, H. F. Time-varying joint distribution through copulas. **Computational Statistics & Data Analysis**, v. 54, n. 11, p. 2383–2399, November 2010. Available from Internet: <https://ideas.repec.org/a/eee/csdana/v54y2010i11p2383-2399.html>.

BACON, C. **Practical Portfolio Performance Measurement and Attribution**. Wiley, 2008. (The Wiley Finance Series). ISBN 9780470778050. Available from Internet: <https://books.google.com.br/books?id=exITHIfwuyUC>.

BLACK, F.; LITTERMAN, R. Global portfolio optimization. **Financial Analysts Journal**, Routledge, v. 48, n. 5, p. 28–43, 1992. Available from Internet: <https://doi.org/10.2469/faj.v48.n5.28>.

CHEKHLOV, A.; URYASEV, S.; ZABARANKIN, M. Portfolio Optimization With Drawdown Constraints. In: **Supply Chain And Finance**. World Scientific Publishing Co. Pte. Ltd., 2004, (World Scientific Book Chapters). chp. 13, p. 209–228. Available from Internet: <https://ideas.repec.org/h/wsi/wschap/9789812562586_0013.html>.

CHERUBINI, U.; LUCIANO, E.; VECCHIATO, W. **Copula Methods in Finance**. Wiley, 2004. (The Wiley Finance Series). ISBN 9780470863459. Available from Internet: <https://books.google.com.br/books?id=0dyagVg20XQC>.

FAN, Y.; PATTON, A. J. Copulas in Econometrics. **Annual Review of Economics**, v. 6, n. 1, p. 179–200, August 2014. Available from Internet: <https://ideas.repec.org/a/anr/reveco/v6y2014p179-200.html>.

HANSEN, P. R. A test for superior predictive ability. **Journal of Business Economic Statistics**, [American Statistical Association, Taylor Francis, Ltd.], v. 23, n. 4, p. 365–380, 2005. ISSN 07350015. Available from Internet: <http://www.jstor.org/stable/27638834>.

HOFERT, M. et al. **copula: Multivariate Dependence with Copulas**. [S.l.], 2018. R package version 0.999-19.1. Available from Internet: <https://CRAN.R-project.org/package=copula>.

HOFERT, M. et al. **Elements of Copula Modeling with R**. Springer Use R! Series, 2018. ISBN 978-3-319-89635-9. Available from Internet: <http://www.springer.com/de/book/9783319896342>.

HU, L. Dependence patterns across financial markets: a mixed copula approach. **Applied Financial Economics**, v. 16, n. 10, p. 717–729, 2006. Available from Internet: <https://EconPapers.repec.org/RePEc:taf:apfiec:v:16:y:2006:i:10:p:717-729>.

KAKOURIS, I.; RUSTEM, B. Robust portfolio optimization with copulas. **European Journal of Operational Research**, v. 235, n. 1, p. 28–37, 2014. Available from Internet: <https://doi.org/10.1016/j.ejor.2013.12.022>.

MARKOWITZ, H. Portfolio selection. **Journal of Finance**, v. 7, n. 1, p. 77–91, 1952. Available from Internet: <https://EconPapers.repec.org/RePEc:bla:jfinan:v:7:y:1952:i:1:p:77-91>.

MORGAN, J. et al. **RiskMetrics: Technical Document**. J. P. Morgan, 1996. Available from Internet: <https://books.google.com.br/books?id=Jm66tgAACAAJ>.

NELSEN, R. An introduction to copulas. **Technometrics**, v. 42, 08 2000.

NELSEN, R. B. **An introduction to copulas.** New York: Springer, 2006. ISBN 0387286594 9780387286594. Available from Internet: <http://www.worldcat.org/search?qt=worldcat_org_all&q=0387286594>.

PATTON, A. J. **Copula-Based Models for Financial Time Series**. [S.l.], 2008. Available from Internet: <https://ideas.repec.org/p/sbs/wpsefe/2008fe21.html>.

PETERSON, B. G.; CARL, P. **PerformanceAnalytics: Econometric Tools for Performance and Risk Analysis**. [S.l.], 2019. R package version 1.5.3. Available from Internet: <https://CRAN.R-project.org/package=PerformanceAnalytics>.

PFAFF, B. **Financial Risk Modelling and Portfolio Optimization with R**. [S.l.: s.n.], 2012. 1-356 p.

PFLUG, G. C. Some remarks on the value-at-risk and the conditional value-at-risk. In: ____. **Probabilistic Constrained Optimization: Methodology and Applications**. Boston, MA: Springer US, 2000. p. 272–281. ISBN 978-1-4757-3150-7. Available from Internet: <https://doi.org/10.1007/978-1-4757-3150-7_15>.

ROCKAFELLAR, R. T.; URYASEV, S. Optimization of conditional value-at-risk. **Journal of Risk**, v. 2, p. 21–41, 2000.

ROCKAFELLAR, R. T.; URYASEV, S. Conditional value-at-risk for general loss distributions. **Journal of Banking and Finance**, p. 1443–1471, 2002.

SILVA, F. B. S. da; ZIEGELMANN, F. Robust portfolio optimization with multivariate copulas: A worst-case cvar approach. 22 2017. Available from Internet: <https://ssrn.com/abstract=3076283>.

SKLAR, M. **Fonctions de Répartition À N Dimensions Et Leurs Marges**. Université Paris 8, 1959. Available from Internet: <https://books.google.com.br/books?id=nreSmAEACAAJ>.

SORTINO, F. A.; MEER, R. van der. Downside risk. **The Journal of Portfolio Management**, Institutional Investor Journals Umbrella, v. 17, n. 4, p. 27–31, 1991. ISSN 0095-4918. Available from Internet: <https://jpm.pm-research.com/content/17/4/27>.

SZEGO, G. Measures of risk. **European Journal of Operational Research**, v. 163, p. 5–19, 05 2005.

WUERTZ, D. et al. **Portfolio Optimization with R/Rmetrics**. [S.l.]: Rmetrics Association Finance Online, www.rmetrics.org, 2010. R package version 2130.80.

XI, L. M. **Portfolio Optimization with PCC-GARCH-CVaR model**. The University of Bergen, 2014. Available from Internet: <https://bora.uib.no/handle/1956/8555>.

YE, Y. **Interior Algorithms for Linear, Quadratic, and Linearly Constrained Non-Linear Programming**. Thesis (Ph.D.) — Department of EES, Stanford University, Stanford, CA, 1987.

ZHU, S.; FUKUSHIMA, M. Worst-case conditional value-at-risk with application to robust portfolio management. **Operations Research**, v. 57, n. 5, p. 1155–1168, 2009. Available from Internet: <https://doi.org/10.1287/opre.1080.0684>.

6 APPENDIX

6.1 Arma-Garch Fitting Code

```
library(xts)
library(quantmod)
library(rugarch)
library(forecast)
setwd("C:/Users/gusta/Desktop/tccGustavo/tccGustavo")


##########load data
returns <- read.csv("returnsIndBr.csv")
returns <- returns[1:5809,]

plot(returns[,3])


####creating a window for each period of opt
per_est = 1260 #number of days used in each estimation
rolling = 1  #you move one observation forward
i=2
window_sup <- 0
window_sup[1]=0
while(window_sup[i-1] < nrow(returns)){
  if(i==2){
    window_sup[i]=per_est
  }
  else{
    window_sup[i]=window_sup[i-1]+rolling
  }
  i=i+1
}
window_sup <- window_sup[-1]
#tail(window_sup)
#window_sup
length(window_sup)
######

#############ARMA GARCH ESTIMATION
```

```
#list of lists to save data
residuos<- vector('list', length(window_sup))
coeficientes <- vector('list', length(window_sup))
sigma <- vector('list', length(window_sup))
aux_list<-vector('list', 9)

for (i in 1:length(window_sup)){ #period 1 to (L-T)
  if(i==1){
    RZ <- returns[1:window_sup[i],]
  }
  else{
    RZ <- returns[(window_sup[i]-per_est+1):window_sup[i],] #goes from
      ↪ window_sup-per_est to window_sup
  }
  for (j in 2:ncol(RZ)) #for each period fits every column (indexes)
  {
    xx <- forecast::auto.arima(y=RZ[,2], max.p=2, max.d = 0, max.q=2,
      ↪ seasonal=F, stationary = T, ic = c('bic'), allowmean =F,
      ↪ allowdrift = F)
    ###max.d=0: wont diff the serie. max(p,q) = 2
    ordem <- forecast::arimaorder(xx) #getting ARMA order
    ordem <- c(ordem[1], ordem[3])

    ##fiting ARMA(p,q)-GARCH(1,1) with skewed t
    armagarchspec <- rugarch::ugarchspec(list(model="sGARCH",
      ↪ garchOrder = c(1,1), variance.targeting = T), mean.model=
      ↪ list(armaOrder=ordem,include.mean = F), distribution.model =
      ↪  'sstd')
    garch_fit <- rugarch::ugarchfit(armagarchspec, data = RZ[,j],solver
      ↪  = "hybrid")

    #storing data
    residuos[[i]][[j-1]] = garch_fit@fit$residuals

    #creating an easy-to-deal matrix of coeficients
    #in the form: (AR1, AR2, MA1, MA2, GARCH coefs)
    if(ordem[1] == 0 && ordem[2] == 0){
      aux_list[1:4] = 0
      aux_list[5:9] = garch_fit@fit$coef[1:5]
    }else
      if(ordem[1] == 1 && ordem[2] == 0){
```

```
    aux_list[2:4]=0
    aux_list[1]=garch_fit@fit$coef[1]
    aux_list[5:9]= garch_fit@fit$coef[2:6]
  } else
    if(ordem[1] == 2 && ordem[2] == 0){
      aux_list[3:4]=0
      aux_list[1:2]=garch_fit@fit$coef[1:2]
      aux_list[5:9]= garch_fit@fit$coef[3:7]
    }
      else
        if(ordem[1]==2 && ordem[2]==1){
          aux_list[4]=0
          aux_list[1:3]=garch_fit@fit$coef[1:3]
          aux_list[5:9]= garch_fit@fit$coef[4:8]
        }
          else
            if(ordem[1]==2 && ordem[2] ==2){
              aux_list[1:4]=garch_fit@fit$coef[1:4]
              aux_list[5:9]= garch_fit@fit$coef[5:9]
            }
              else
                if(ordem[1]==0 && ordem[2]==1){
                  aux_list[1:2]=0
                  aux_list[4]=0
                  aux_list[3]=garch_fit@fit$coef[1]
                  aux_list[5:9]= garch_fit@fit$coef[2:6]
                }
                  else
                    if(ordem[1]==0 && ordem[2]==2){
                      aux_list[1:2]=0
                      aux_list[3:4]=garch_fit@fit$coef[1:2]
                      aux_list[5:9]= garch_fit@fit$coef[3:7]
                    }
                      else
                        if(ordem[1]==1 && ordem[2] ==2){
                          aux_list[2]=0
                          aux_list[1]=garch_fit@fit$coef[1]
                          aux_list[3:4]=garch_fit@fit$coef[2:3]
                          aux_list[5:9]= garch_fit@fit$coef[4:8]
                        } else
                        if(ordem[1]==1 && ordem[2]==1){
```

```
                              aux_list[2]=aux_list[4]=0
                              aux_list[1]=garch_fit@fit$coef[1]
                              aux_list[3]=garch_fit@fit$coef[2]
                              aux_list[5:9]=garch_fit@fit$coef[3:7]
                              }
    coeficientes[[i]][[j-1]] = aux_list
    sigma[[i]][[j-1]] = garch_fit@fit$sigma
    cat("\nFit␣Col/Per:␣",j,i)
  }
}

#################saving data in R file
saveRDS(coeficientes, file = "coef.Rds")
saveRDS(residuos, file = "residuos.Rds")
saveRDS(sigma, file = "sigma.Rds")

#######generating parametric cdf with uniform margins
residuos <- readRDS("residuos.Rds") ##reading our data
sigma <- readRDS("sigma.Rds")
garch_coef <- readRDS("coef.Rds")

##initializing an empty list to save data
unif <- vector('list', 4550 )

##generating Cdf with a parametric skewd t, given degrees of freedom
   ↪ and skewness
for(i in 1:length(residuos)){
  for(j in 1:8){
    unif[[i]][[j]] = fGarch::psstd(q=residuos[[i]][[j]]/sigma[[i]][[j
       ↪ ]],
                  nu = garch_coef[[i]][[j]][[8]],
                  xi = garch_coef[[i]][[j]][[7]])
  }
}
####saving ecdf data
saveRDS(unif, file = "Unif_ParDist.Rds")
```

6.2 Mixture Copula Parameter Fit Code

```r
library(copula)   #for estimating copulae
library(Rsolnp)   #for non linear optimization

setwd("C:/Users/gusta/Desktop/tccGustavo/tccGustavo")

#X <- readRDS("unif_EmpDist.RDS")    #reading our empirical cumulative unif. distr.
X <- readRDS("unif_ParDist.RDS") #reading our parametric cdf

weight_t <- vector('list', length(X) )  #creating a list to store each period's parameters

#negative log likelihood function for estimating copulae weights and parameters
LLCG <- function(params,U, copC, copG, copt){
  slot(copC, "parameters") <- params[1]
  slot(copG, "parameters") <- params[2]
  slot(copt, "parameters") <- params[3:4]
  pi1 <- params[5]
  pi2 <- params[6]
  pi3 <- params[7]

  opt <- log(pi1 * dCopula(U, copC) + pi2 * dCopula(U, copG)
             + pi3 * dCopula(U, copt))
  if(any(is.infinite(opt))){
    opt[which(is.infinite(opt))]<-0
  }
  -sum(opt)
}

#constrain function so that sum(weights)=1
eqfun <- function(params,U,copC,copG,copt){
  z <- params[5]+params[6]+params[7]
  return(z)
}

#initializing 8d t copula
copt <- copula::tCopula(param = 0.5, dim = 8) #

#initializing archimedian copula objects
copC <- copula::claytonCopula(2, dim = 8) # delta= 2
```

```
copG <- copula::gumbelCopula(2, dim = 8)  # theta= 2


#lower and upper bounds of the parameters and weights
lower <- c(0.1, 1, -0.9,(2+.Machine$double.eps), 0,0,0)
upper <- c(copC@param.upbnd, copG@param.upbnd, 1,100, 1,1,1) #2+eps so
    ↪ that variance of t copula is defined



for(i in i:length(X)){ #for i in 1:4550

##unif dataframe
v<-as.matrix(do.call(cbind, X[[i]]))
U<-v[,1:8]

## Creating elliptical copula objects
par1 <- copula::fitCopula(copC, U, "itau",estimate.variance= T)
    ↪ @estimate #inversion of Kendall's tau
par2 <- copula::fitCopula(copG, U,"itau", estimate.variance= T)
    ↪ @estimate
par3 <- copula::fitCopula(copt, U,"mpl",estimate.variance= F)@estimate
    ↪ ###you need to use mpl in order to estime Degrees of freedom as
    ↪ well
par4 <- 1/3
par5 <- 1/3
par6 <- 1/3

##non linear constrained optimization
opt <- Rsolnp::solnp(pars = c(par1,par2,par3,par4,par5,par6), fun =
    ↪ LLCG, LB = lower,
                     UB = upper, copt=copt,copC = copC, copG = copG, U=
                        ↪ U,eqfun = eqfun,
                     eqB=c(1)) ####RSOLNP

weight_t[[i]]<-opt$pars  ##saving parameters in a list
}


saveRDS(weight_t, file = "copulaParams.Rds")  #saving the parameters in
    ↪  a file
```

6.3 Gauss Copula Parameter Fit

```
library(copula)   #for estimating copulae

setwd("C:/Users/gusta/Desktop/tccGustavo/tccGustavo")

#X <- readRDS("unif_EmpDist.RDS")    #reading our empirical cumulative
    ↪ unif. distr.
X <- readRDS("unif_ParDist.RDS") #reading our parametric cdf

weight_gauss<- vector('list', length(X))

copGauss <- copula::normalCopula(dim = 8)

for(i in i:length(X)){

  ##unif dataframe
  v<-as.matrix(do.call(cbind, X[[i]]))
  U<-v[,1:8]

  ## Creating elliptical gaussian copula object
  opt <- copula::fitCopula(copGauss, U, "itau")@estimate #inversion of
      ↪ Kendall's tau

  weight_gauss[[i]]<-opt  ##saving parameters in a list
}

saveRDS(weight_gauss, file = "copulaGauss.Rds")
```

6.4 Mixture Copula Scenario Generation and Portfolio Optimization Code

```
library(copula) ##for generating copula observations
library(fGarch) ##for using skew t quantile function
library(fPortfolio) ##portfolio optimization library


cop_pars <- readRDS("copulaParams.Rds")  #reading copula par
garch_coefs <- readRDS("coef.Rds") #reading ArmaGarch par
```

```
arma_order <- readRDS("armaOrder.Rds")
sigma_fit <- readRDS("sigma.Rds")          ##reading armaGarch fitted
    ↪ residuals and sigma. We need this to estimate J one-steap ahead
    ↪ returns
residual_fit <- readRDS("residuos.Rds")

returns <- read.csv("returnsIndBr.csv") #reading return data
returns <- returns[1:5809,] #this is all data we use


nsim = 10000 #K number of scenarios

##initializing needed matrix
cvar_opt <- matrix(0, nrow= 4550, ncol=9)
Cc <- Cg <- Ct <- matrix(0,nrow = 10000, ncol = 8)
ctg <- matrix(0,nrow = 10000, ncol = 8)
zsim <- matrix(0,nrow = 10000, ncol = 8)
ret_ <- matrix(0, nrow=10000, ncol = 8)

##setting up a fGarch portfolio
frontierSpec <- portfolioSpec()
setType(frontierSpec) <-"CVaR"
setSolver(frontierSpec)  <- "solveRglpk.CVAR"
setAlpha(frontierSpec) <-0.05
setTargetReturn(frontierSpec) <- 0.00012  #we do this for 0, 0.00012,
    ↪ 0.00024 and 0.00036


for (i in 1:length(sigma_fit)) {
    ret_ <- matrix(0, nrow=10000, ncol = 8)

    ##generating simulated copula data
    Cc[,]<- cop_pars[[i]][[5]]*copula::rCopula(n = 10000, copula =
        ↪ claytonCopula(param = cop_pars[[i]][[1]],dim= 8))
    Cg[,]<- cop_pars[[i]][[6]]*copula::rCopula(n = 10000, copula =
        ↪ gumbelCopula(param = cop_pars[[i]][[2]],dim= 8))
    Ct[,]<- cop_pars[[i]][[7]]*copula::rCopula(n = 10000, copula =
        ↪ tCopula(param = cop_pars[[i]][[3]], df=cop_pars[[i]][[4]],
        ↪ dim= 8))

     ctg <- Cc + Ct + Cg #linear combination of copula
```

```
  ##calculating Zsim for each j asset every i day
  for(j in 1:8){
    zsim[,j] <- fGarch::qsstd(ctg[,j], nu = garch_coefs[[i]][[j
        ↪ ]][[8]], xi =garch_coefs[[i]][[j]][[7]]) /
              sd(fGarch::qsstd(ctg[,j], nu = garch_coefs[[i]][[j
                 ↪ ]][[8]], xi = garch_coefs[[i]][[j]][[7]]))
              #nu = DF da t, xi = skew da t
  }

##K scenarios returns simulation using Zsim and ARMA-GARCH

#matrix of real returns for AR term
RZ<-returns[i:(1259+i),2:9]

#sigma and residual matrix for MA and GARCH
sigma_per <- sigma_fit[[i]]
resid_per <- residual_fit[[i]]

##generating K returns scenarios for each asset
for(j in 1:8){
   sigma_f_t1 <- tail(sigma_per[[j]],1) ##(t-1)
   e_f_t2_t1 <- tail(resid_per[[j]],2) ##(t-2, t-1)
   for(z in 1:10000){
       ret_[z,j] = ((garch_coefs[[i]][[j]][[1]] * RZ[1260,j]) + (
          ↪ garch_coefs[[i]][[j]][[2]] * RZ[1259,j]) +  #AR1 * R_t
          ↪ -1, AR2 * R_t-2
                   (garch_coefs[[i]][[j]][[3]] * e_f_t2_t1[1]) + (
                      ↪ garch_coefs[[i]][[j]][[4]] * e_f_t2_t1[2])
                      ↪  +  #MA1*e_t-1, MA2*e_t-2
                   (zsim[z,j] * (sqrt(garch_coefs[[i]][[j]][[9]]) +
                      ↪  ##alfa0
                                sqrt(garch_coefs[[i]][[j]][[5]]) *
                                   ↪ e_f_t2_t1[2] + ##alfa1 * e_t
                                   ↪ -1
                                sqrt(garch_coefs[[i]][[j]][[6]]) *
                                   ↪ sigma_f_t1))) ##beta1  * s_t
                                   ↪ -1
                 }
}
```

```
  ##optimizing portfolio for period i in 1:4550 using simulated returns
  retornofPort <- as.timeSeries(ret_[,1:8])
  frontier1g   <- fPortfolio::efficientPortfolio(data = retornofPort ,
     ↪ spec = frontierSpec ,constraints ="LongOnly")

  cvar_opt[i,1:8] <- fPortfolio::getWeights(frontier1g)
  cvar_opt[i,9]   <- fPortfolio::getTargetRisk(frontier1g)[3]

}

saveRDS(cvar_opt, file = "mixW0pct.Rds") ##we repeat this for 0,3,6 and
   ↪  9%
```

6.5 Gauss Copula Scenario Generation and Portfolio Optimization Code

```
cop_pars <- readRDS("copulaGauss.Rds")  #reading gaussian copula par
garch_coefs <- readRDS("coef.Rds") #reading ArmaGarch par
arma_order <- readRDS("armaOrder.Rds")
sigma_fit <- readRDS("sigma.Rds")        ##reading armaGarch fitted
   ↪ residuals and sigma. We need this to estimate J one-steap ahead
   ↪ returns
residual_fit <- readRDS("residuos.Rds")

returns <- read.csv("returnsIndBr.csv")
returns <- returns[1:5809,]



nsim = 10000 #number of scenarios

#initializing matrix
cvar_opt <- matrix(0, nrow= 4550, ncol=9)
Gcop <- matrix(0,nrow = 10000, ncol = 8)
zsim <- matrix(0,nrow = 10000, ncol = 8)
ret_ <- matrix(0, nrow=10000, ncol = 8)

#fPortfolio mean-CVaR spec
frontierSpec <- portfolioSpec()
setType(frontierSpec) <-"CVaR"
setSolver(frontierSpec)  <- "solveRglpk.CVAR"
```

```
setAlpha(frontierSpec) <-0.05
setTargetReturn(frontierSpec) <- 0.00012



for (i in 1:1) {
  ret_ <- matrix(0, nrow=10000, ncol = 8)

  #there is no linear combination of copulas to be generated, only
     ↪ GaussianCopula data
  Gcop[,]<- copula::rCopula(n = 10000, copula = normalCopula(param =
     ↪ cop_pars[[i]],dim= 8))


  for(j in 1:8){
    if(j==8 && anyNA(fGarch::qsstd(ctg[,j], nu = garch_coefs[[i]][[j
       ↪ ]][[8]], xi =garch_coefs[[i]][[j]][[7]]))){
      garch_coefs[[i]][[j]][[7]] <- 1
    }
    zsim[,j] <- fGarch::qsstd(Gcop[,j], nu = garch_coefs[[i]][[j
       ↪ ]][[8]], xi =garch_coefs[[i]][[j]][[7]]) / # garch_coefs[[i
       ↪ ]][[j]][[7]]
      sd(fGarch::qsstd(Gcop[,j], nu = garch_coefs[[i]][[j]][[8]], xi =
         ↪ garch_coefs[[i]][[j]][[7]])) #nu = GL da t, xi = skew da t
  }



  RZ<-returns[i:(1259+i),2:9]
  sigma_per <- sigma_fit[[i]]
  resid_per <- residual_fit[[i]]

  for(j in 1:8){
    sigma_f_t1 <- tail(sigma_per[[j]],1) ##(t-1)
    e_f_t2_t1 <- tail(resid_per[[j]],2) ##(t-2, t-1)
    for(z in 1:10000){
      ret_[z,j] = ((garch_coefs[[i]][[j]][[1]] * RZ[1260,j]) + (garch_
         ↪ coefs[[i]][[j]][[2]] * RZ[1259,j]) +  #AR1 * R_t-1, AR2 *
         ↪ R_t-2
                    (garch_coefs[[i]][[j]][[3]] * e_f_t2_t1[1]) + (
                       ↪ garch_coefs[[i]][[j]][[4]] * e_f_t2_t1[2])
                       ↪ +  #MA1*e_t-1, MA2*e_t-2
```

```
                    (zsim[z,j] * (sqrt(garch_coefs[[i]][[j]][[9]]) +
                      ↪ ##alfa0
                                  sqrt(garch_coefs[[i]][[j]][[5]]) *
                                    ↪  e_f_t2_t1[2] + ##alfa1 * e
                                    ↪ _t-1
                                  sqrt(garch_coefs[[i]][[j]][[6]]) *
                                    ↪  sigma_f_t1))) ##beta1  * s
                                    ↪ _t-1
    }
  }
  retornofPort <- as.timeSeries(ret_[,1:8])
  frontier1g <- efficientPortfolio(data = retornofPort , spec =
     ↪ frontierSpec ,constraints ="LongOnly")

  cvar_opt[i,1:8] <- getWeights(frontier1g)
  cvar_opt[i,9] <- getTargetRisk(frontier1g)[3]

}
```

6.6 Portfolio Benchmark

```
##every function here is from PerformanceAnalytics
library(PerformanceAnalytics)
library(openxlsx)

w_mix_3 <- readRDS("mixW9pct.Rds") ##reading 3% yearly mixture
   ↪ optimized portfolio weights
w_gaus_3 <- readRDS("gaussW9pct.Rds") ##reading 3% yearly gaussian
   ↪ optimizaed portfolio weights

##reading real log returns data-set
returns <- read.csv("returnsIndBr.csv")

##subsetting to keep only the out-of-sample returns we will use
returns <- returns[1261:5810, ,drop=F]


time <- as.Date(returns[,1]) ###converting data-sets date into Date
```

```
returns <- returns[,2:9,drop=F]  ##dropping Date column

simple_ret <- matrix(0,nrow=4550, ncol=8)

####transforming from log-returns to simple-returns
simple_ret <- exp(returns)-1

##transforming weights to xts
aux_date <- as.Date(c("2001-02-05","2001-02-06"))

w_mix_31 <- as.xts(w_mix_3[1:2,1:8],order.by = aux_date)
w_mix_31 <- w_mix_31[1,]
w_mix_3 <- as.xts(w_mix_3[-1,1:8], order.by = time2)
w_mix_3 <- rbind(w_mix_31, w_mix_3)

simple_ret <- as.xts(simple_ret, order.by = time)

w_gaus_31 <- as.xts(w_gaus_3[1:2,1:8],order.by = aux_date)
w_gaus_31 <- w_gaus_31[1,]
w_gaus_3 <- as.xts(w_gaus_3[-1,1:8], order.by = time2)
w_gaus_3 <- rbind(w_gaus_31, w_gaus_3)

#w_markowitz_3 <- as.xts(w_markowitz_3[,1:8], order.by = time)
#w_markowitz_3_shrink <- as.xts(w_markowitz_3_shrink[,1:8], order.by =
    ↪ time)

##assingning names to xts matrix (needed for return calculation using
    ↪ Return.portfolio)
colnames(simple_ret) <- colnames(w_mix_3) <- colnames (w_gaus_3) <- c("
    ↪ a","b","c","d","e","f","g","h")
#colnames(w_markowitz_3) <- colnames(w_markowitz_3_shrink)

colnames(simple_ret) <- colnames(w_gaus_3) <- c("a","b","c","d","e","f"
    ↪ ,"g","h")
#####out of sample returns (i=1)
outSamRet_3_ctg <- Return.portfolio(simple_ret, w_mix_3, geometric = T,
    ↪  wealth.index = F)
outSamRet_3_gaus <- Return.portfolio(simple_ret, w_gaus_3, geometric =
    ↪ T, wealth.index = F)
outSamRet_3_1n  <- Return.portfolio(simple_ret, geometric = T, wealth.
    ↪ index = F)
```

```
outSamRet_3_IBOV <- Return.portfolio(simple_ret[,4], geometric = T) ##
    ↪ simple_ret[,4] = ibov

returnsmatrix9 <- cbind(outSamRet_3_ctg, outSamRet_3_gaus, outSamRet_3_
    ↪ 1n,outSamRet_3_IBOV) # creating a matrix of portfolios returns
colnames(returnsmatrix9) <- c("Ctg", "Gauss", "1n", "IBOV")



##########performance measures and plots
charts.PerformanceSummary(returnsmatrix)

#chart.Drawdown(returnsmatrix, legend.loc = "bottomright", colorset =
    ↪ rainbow6equal)

###Risk-Free rate and MAR = 0
bench<-NULL
bench <- rbind(Return.annualized(returnsmatrix), sd.annualized(
    ↪ returnsmatrix),
               VaR(returnsmatrix, method = "historical"), CVaR(
                   ↪ returnsmatrix, method = "historical"),
               SemiDeviation(returnsmatrix), CDD(returnsmatrix, method
                   ↪ = "historical"),
               maxDrawdown(returnsmatrix, method = "historical"),
               AverageDrawdown(returnsmatrix, method = "historical"),
                   ↪ AverageLength(returnsmatrix, method = "historical
                   ↪ "),
               SharpeRatio.annualized(returnsmatrix, Rf = 0),
               BurkeRatio(returnsmatrix, Rf = 0),
               SortinoRatio(returnsmatrix, MAR = 0),
                   ↪ UpsidePotentialRatio(returnsmatrix, MAR = 0),
               DownsideFrequency(returnsmatrix, MAR = 0)
              , CalmarRatio(returnsmatrix),
               DrawdownDeviation(returnsmatrix),
               Kappa(returnsmatrix,MAR=0, 1),
               OmegaSharpeRatio(returnsmatrix, MAR=0))

write.xlsx(bench, file = "benchmark6pct.xlsx", colNames = T, rowNames =
    ↪  T) ##saving performance measures



###drawdowns plot
```

```
chart.Drawdown(R=returnsmatrix0, legend.loc = "bottomright", colorset =
    ↪  rainbow6equal, main = "0.000000")
chart.Drawdown(returnsmatrix3, legend.loc = "bottomright", colorset =
    ↪ rainbow6equal,main = "0.000120")
chart.Drawdown(returnsmatrix6, legend.loc = "bottomright", colorset =
    ↪ rainbow6equal,main = "0.000240")
chart.Drawdown(returnsmatrix9, legend.loc = "bottomright", colorset =
    ↪ rainbow6equal,main = "0.000360")
```