Trabalho II

Bruno Grillo bruno.grillo@ufrgs.br
193827

Obs: todos scripts estão na última seção

1. Introdução

O presente trabalho tem como objetivo comparar o desempenho de diferentes estratégias de alocação de portfólio para um conjunto limitado de ativos. As séries contém os preços de fechamento diários entre 01/01/2016 e 31/12/2020 de ações de seis companhias listadas na B3: Lojas Renner SA (LREN3), Localiza (RENT3), Magazine Luiza (MGLU3), WEG (WEGE3), Banco do Brasil (BBAS3) e Embraer (EMBR3). Os dois últimos anos de observações, 01/01/2019 a 31/12/2020, são usados como período de avaliação das estratégias distintas. A abordagem candidata de alocação de portfólio se dá pela otimização da medida de risco CVaR para níveis distintos de rentabilidade usando cópulas para estimar dependência entre ativos e ARMA-GARCH para modelar média e volatilidade. A frequência de atualização do portóflio é diária, usando os dados disponíveis do dia anterior. É importante ressaltar que considera-se o custo de transação igual a zero, ou seja, não incorpora-se na otimização os custos em vender/comprar ativos.

Para dados econômicos, é muito comum encontrar variáveis que não seguem distribuição normal. Há algumas características frequentemente observadas em séries de retornos diários de ativos financeiros, que são denominadas ”fatos estilizados”. Em Pfaff (2016) são elecandos alguns desses padrões presentes em dados de mercados financeiros. Essas propriedades têm implicações importantes na hora de avaliar se um modelo é apropriado ou não, visto que o modelo não será útil para derivar medidas de risco se ele falha em capturar essas características adequadamente. Os fatos estilizados são divididos em dois grupos, de séries univariadas e de séries multivariadas. Em relação às séries univariadas tem-se que **(i)** as séries temporais de retornos diários não são, no geral, independente- e identicamente distribuídas, fato que não é comprometido por baixos valores absolutos do coeficiente de autocorrelação de primeira-ordem, **(ii)** a volatilidade do processo de retornos não é contante em relação ao tempo, **(iii)** os retornos absolutos e quadráticos são altamente autocorrelacionados, **(iv)** a distribuição tende a ser leptocúrtica, ou seja, a ocorrência de eventos extremos é mais provável que comparada a uma normal (cauda pesada), **(v)** retornos extremos são observados com proximidade (clusters de volatilidade), e **(vi)** a distribuição empírica dos retornos é assimétrica à esquerda, ou seja, há uma frequência maior de retornos negativos do que positivos. Já para as séries multivariadas tem-se **(i)** correlações contemporâneas não são constantes ao longo do tempo, e **(ii)** observações extremas em uma série de retornos é frequentemente acompanhada por extremos em outras séries de retornos. Algo que é verificado nos dados para o período de início da pandemia do Covid-19, onde todos ativos tiveram movimentos similares devido às incertezas geradas em escala global.

Uma forma de lidar com as características supracitadas é com o uso de cópulas, que é útil para modelar relação de dependência entre variáveis. Nesse trabalho, testa-se o uso de três cópulas distintas (Gumbel, t-Student e Joe) e uma mistura entre as cópulas. Hofert (2018) sugere que no contexto de cópulas estimadas não serem diferentes da 'cópula independente' apenas a estimativa das marginais é necessária, usando métodos estatísticos clássicos para séries univariadas.

Em seu trabalho seminal denominado Teoria Moderna de Portfólio, Markowitz (1952) faz um discernimento revolucionário sobre a relação retorno/risco, ou média/variância. O autor sugere que não deve-se avaliar a relação para cada ativo separadamente, mas no contexto de portfólio. O autor elabora um framework para maximizar o retorno de um portfólio para um determinado risco. Para um portfólio ser considerado eficiente ele precisa ou (i) ter o maior retorno para um determinado nível de risco ou (ii) ter o menor nível de risco para um determinado retorno. De acordo com Pfaff (2016), embora as duas visões sejam equivalentes, o processo de otimização do portfólio difere para os dois casos. O primeiro é uma otimização quadrática com restrições lineares e o segundo é uma otimização linear com restrições quadráticas.

No contexto de economia, classifica-se os agentes em grupos a partir do grau de aversão ao risco, que são: (i) avesso ao risco, que prefere resultados com menores incertezas, mesmo que com retorno esperado menor; (ii) neutro ao risco, que prefere o máximo valor esperado independente do risco associado, e (iii) propensos ao risco, que são agentes que possuem funções de utilidade convexas, ou seja, utilidade crescente a taxas crescentes em função da recompensa. Pesquisas mais recentes de economia comportamental trazem novos elementos em relação à tomada de decisão de agentes frente ao risco. Um desses conceitos, o de aversão a perda, identificado por Kahneman e Tversky (1979), sugere que os efeitos psicológicos das perdas são maiores que dos ganhos.

O risco trazido por Markowitz inclui tanto o risco de ganhar (a parte boa) quanto o de perder dinheiro. Na literatura de finanças, há métricas para lidar com o *downside risk*, que representa o risco do retorno ser inferior ao esperado, e otimizar a alocação de ativos num portfólio a partir delas. Nesse contexto, uma medida de *downside risk* amplamente utilizada é o VaR (*value at risk*), contudo, Pfaff (2016) aborda algumas limitações dela, entre elas (i) não-convexidade, que pode levar a ótimos locais, (ii) viola a subaditividade, propriedade que sugere que o risco de portfólio é menor ou igual à soma do risco individual de cada ativo contido no portfólio, e (iii) não transmite informação adequada sobre as perdas potenciais de cauda. CVaR, ou *Expected Shorfall* (ES), é uma medida de risco que contorna essas limitações.

2. Método

Nessa seção descreve-se a abordagem adotada para otimizar portfólio de acordo com CVaR.

2.1 ARMA-GARCH

Emprega-se o algoritmo desenvolvido por Hyndman (2008) para determinar as ordens do modelo ARMA para a média do retorno. Restringe-se a busca em modelos ARMA(p,q) não-sazonais e sem o parâmetro de média, pois a série de log-retornos diários é estacionária com média aproximadamente igual a zero. Essa estratégia de seleção de modelo foi implementada na função *auto.arima()* do pacote *forecast*. Um parâmetro fundamental dessa implementação é o *max.order*, que determina o número máximo que a soma de ordens p, q pode atingir. Para esse trabalho, emprega-se *max.order* = 8.

Utiliza-se o modelo GARCH para estimar a volatilidade, as funções usadas para especificação e estimação são, respectivamente, *ugarchfit* e *ugarchspec* do pacote *rugarch*. Restringe-se a ordem GARCH para (1,1) e as inovações seguem uma distribuição t-Student assimétrica padronizada em virtude dos fatos estilizados elencados na introdução. Ainda em relação à parametrização do GARCH, mantém-se o intercepto, pois ao forçar o intercepto a ser igual a zero: (i) e a soma dos coeficientes ARCH e GARCH for inferior a um, isso implica em a variância condicional ser decrescente ao longo do tempo; (ii) e a soma dos coeficientes ARCH e GARCH for igual a 1, a variância condicional será um passeio aleatório.

Segue o procedimento descrito em forma de algoritmo, que é aplicado em cada uma das séries univariadas de log-retorno dos ativos. Em outras palavras, para cada ativo há um modelo específico.

Algorithm 1 Determinar ARMA(p,q)-GARCH(1,1)

```
procedure ARMA_GARCH(y)                                      ▷ y = regredida
    Restringe-se a busca por modelos estacionários e sem sazonalidade
    Gera-se todas combinações possíveis de ordens a testar
    for cada combinação de ordens do
        Estima-se modelo ARMA sem intercepto
        Computa-se o critério de informação de Akaike (AIC)
    end for
    Compara-se AIC dos modelos candidatos
    Seleciona-se (p,q) do modelo com menor AIC
    Estima-se GARCH com:
              média (p,q)
              variância (1,1)
              inovações t-Student assimétrica
    return modelo_final
end procedure
```

2.2 Cópulas

Hofert (2018) define cópula como função de distribuição multivariada com marginais univariadas uniformes-padrão $U[0,1]$ (contínuas). O autor ressalta, entretanto, que a escolha de marginais uniforme-padrão pode ser vista arbitrária, pois a forma de padronização não altera a filosofia das cópulas. Contudo, afirma que é mais natural considerar funções de distribuição cujas marginais sejam Fréchet-unitárias devido às transformações de probabilidade e quantílicas.
Pelo teorema de Sklar (1959), cópulas são funções que combinam as distribuições marginais univariadas para formar funções de distribuição de qualquer dimensão. Pelo teorema, toda função de distribuição d-dimensional H pode ser expressa em termos de uma cópula d-dimensional C e suas funções de distribuição marginal $F_1, ..., F_d$ obtidas de H.

$$H(\mathbf{x}) = C(F_1(x_1), ..., F_d(x_d)),\ \mathbf{x} \in R^d$$

Ou seja, elas conectam funções de distribuição multivariadas H com as suas marginais univariadas $F_1, ..., F_d$. Elas possuem informações acerca da estrutura de dependência entre as variáveis, o que permite uma maior flexibilidade ao modelar as distribuições multivariadas e suas marginais, pois oferece mais detalhes nas especificações ao permitir modelar separaradamente as características específicas de cada marginal do parâmetro de dependência representado pela C.

Empregou-se três cópulas distintas e mais uma mistura delas. As três cópulas são:

1. *t-Student* **Copula**: com parâmetros ρ e *df* e pertencente à classe de cópulas elípticas

2. **Gumbel Copula**: com parâmetro α_G e pertencente à classe de cópulas arquimedianas (e de valores extremos).

3. **Joe Copula**: com parâmetro α_J pertencente à classe de cópulas arquimedianas

4. **Mistura de Cópulas** com parâmetros $\alpha_G, \rho, df, \alpha_J$ e pesos w_G, w_t, w_J.

Nessa aplicação, deseja-se utilizar apenas uma cópula para modelar a estrutura de dependência, e é escolhida pelo critério de informação de Akaike (AIC). Para a estimação das cópulas por máxima verossimilhança, emprega-se a função *fitCopula* implementada no pacote *copula*. O procedimento para escolher a cópula segue os seguintes passos:

É importante ressaltar que o critério de informação não diz nada em relação à qualidade

Algorithm 2 Determinar Copula

```
procedure COPULAE_FIT(y, goft = FALSE)  ▷ y = pseudo_observations, goft = teste qualidade de ajuste
    Estima-se cópula de valor extremo e arquimediana Gumbel para y (parâmetro α)
    Estima-se cópula elíptica t-Student para y (parâmetros ρ e df)
    Estima-se cópula arquimediana Joe para y (parâmetro α
    Estima-se mistura das cópulas acima (parâmetros α_G, ρ, df, α_J e pesos w_G, w_t, w_J)
    for cada cópula do
        Computa-se o critério de informação de Akaike (AIC
        Computa-se o critério de informação de Bayesiano (BIC)
        Computa-se a máxima verossimilhança
    end for
    if goft = TRUE then
        Computa-se teste de qualidade de ajuste para cada cópula
        Seleciona-se a cópula com menor AIC (pode-se mudar esse parâmetro)
        resultado ← (copula_final, teste_GoF)
    else
        Seleciona-se a cópula com menor AIC (pode-se mudar esse parâmetro)
        resultado ← copula_final
    end if
    return resultado
end procedure
```

de ajuste. Quando o parâmetro *goft* do procedimento acima não é o *default* e utiliza-se *goft = TRUE*, emprega-se o teste de qualidade de ajuste proposto por Genest *et al* (2009), que consiste no critério de Cramér–von Mises e a estatística de teste é calculada através de *bootstrap* paramétrico, o que torna o cálculo extremamente lento.

2.3 Otimização CVaR

Emprega-se um critério para gerar portfólio otimizados em relação ao risco, o portfólio escolhido será o resultado da otimização pelo critério do CVaR, também conhecido como *tail*-VaR ou *Expected Shorfall* (ES). Essa métrica de risco é definida como a perda esperada excedendo o VaR para um determinado nível de confiança. Pfaff(2016) ainda referencia trabalhos dos autores Rockafellar and Uryasev demonstrando que VaR não é uma medida de risco coerente, visto que pode ser não-convexa e gerar, portanto, soluções ótimas localmente (e não globalmente).

Pfaff (2016) define $CVaR$ como a média ponderada entre VaR e $CVaR^+$ em determinado nível de confiança, onde $CVaR^+$ é a perda esperada estritamente excedente ao VaR

$$CVaR = \lambda VaR + (1 - \lambda) CVaR^+$$

Onde o peso $0 \leqslant \lambda \leqslant 1$ é dado por $\lambda = \frac{\Psi(VaR)-\alpha}{1-\alpha}$ e $\Psi(VaR)$ representa a probabilidade que as perdas não excedam ou sejam iguais ao VaR para um determinado nível de confiança.

Um problema que surge na otimização do CVaR é a dependência do VaR, que pode ser não-convexo. Os autores citados por Pfaff (2016) na seção 12.3, Rockafellar and Uryasev, desenvolveram uma função auxiliar sem necessidade de computar o VaR. Para otimização, emprega-se as funções *portfolioSpec* e *efficientPortfolio* implementadas no pacote *fPortfolio* e segue-se sugestão dos autores do pacote em Würtz *et al* (2105) de utilizar programação linear usando o parâmetro $solver = solveRglpk.CVAR$. É importante ressaltar que a otimização de CVaR pode ser formulada como um problema de programação linear se for possível gerar simulações Monte-Carlo nos retornos.

2.4 Procedimento Portfólio

Nessa subseção explica-se o fluxo utilizado para construir cada portfólio. Emprega-se procedimentos descritos em subseções anteriores. Nesse exercício, o nível de confiança utilizado no CVaR é de 95%. Embora o critério de informação não diga nada em relação à qualidade de ajuste, na execução da estratégia empregou-se o AIC como regra de escolha e ignorou-se a significância da cópula, ou seja, podem ter sido utilizadas cópulas não-signicativas. Todavia, na seção de resultados haverá um exemplo de simulação com todas estatísticas de teste.

Na linha 12 do algoritmo 3, a simulação dos log-retorno para cada ativo a no período t é dada por $\mathbf{R}_{t,i} = \hat{\mu}_{t,i} + \hat{\sigma}_{t,i}\mathbf{z}_{t,i}$, onde R representa o retorno, $\hat{\mu}$ a previsão da média pelo modelo ARMA, $\hat{\sigma}$ a previsão da volatilidade pelo modelo GARCH e $\mathbf{z}$ o quantil da t-Student (nesse caso, $\mathbf{z}$ é um vetor de comprimento 1000).

Algorithm 3 Determinar Portfólio

1: **procedure** PORTFOLIO_WEIGHTS($y, target$) ▷ y = matriz de log-retornos, target = taxa retorno alvo
2: Determinar quantidade de ativos
3: **for** cada série de ativo **do**
4: Chamar função $arma\_garch(y = série\_ativo)$
5: Criar vetor de resíduos padronizados
6: Criar vetor com transformação resíduos padronizados em pseudo-uniformes
7: **end for**
8: Criar matriz pseudo-uniforme a partir dos vetores univariados
9: Chamar função $copulae\_fit(y = matriz\_pseudo\_uniforme)$
10: Gerar 1000 simulações a partir da estrutura de dependência da cópula
11: Calcular os quantis da t-Student assimétrica das simulações de Monte Carlo
12: Simular os log-retornos diários para cada ativo (ver comentário acima)
13: Definir pesos minimizando CVaR para *target* a partir dos log-retornos simulados.
14: **return** pesos_portfólio
15: **end procedure**

3. Resultados

Essa seção está dividida em três partes: (i) na primeira parte há uma descrição dos dados, (ii) na segunda verifica-se o resultado da estratégia de otimização de portfólio por cópulas e compara-se com abordagens ingênuas (pesos idênticos entre ativos) com recalibragem diária e mensal, (iii) na terceira parte escolhe-se um período para avaliar os parâmetros gerados e estatísticas de teste.

3.1 Descrição dos Dados

A tabela abaixo mostra algumas estatísticas descritivas para as 1242 observações de log-retornos de cada ativo considerado durante o período inteiro de avaliação 01/01/2016 a 31/12/2020.

Table 1: Estatísticas Descritivas Log-Retorno

	Mean	Median	Min	Max	Std. Deviation	Skewness	Kurtosis
BBAS3	0.001	0.001	-0.238	0.158	0.030	-0.728	13.549
EMBR3	-0.001	0.000	-0.307	0.203	0.030	-0.893	19.920
LREN3	0.001	0.001	-0.237	0.153	0.025	-0.695	14.663
MGLU3	0.005	0.003	-0.237	0.269	0.039	0.083	10.464
RENT3	0.002	0.001	-0.269	0.238	0.029	-0.331	16.597
WEGE3	0.002	0.002	-0.231	0.130	0.023	-0.975	16.411

Pela tabela acima, observa-se uma particularidade nos retornos do ativo MGLU3, que possui assimetria à direita e apresenta o menor valor de curtose. Na imagem abaixo está a evolução temporal, onde observa-se um cluster de volatilidade por volta de março de 2020, período inicial da pandemia Covid-19. É importante ressaltar que os eixos estão exatamente na mesma escala.

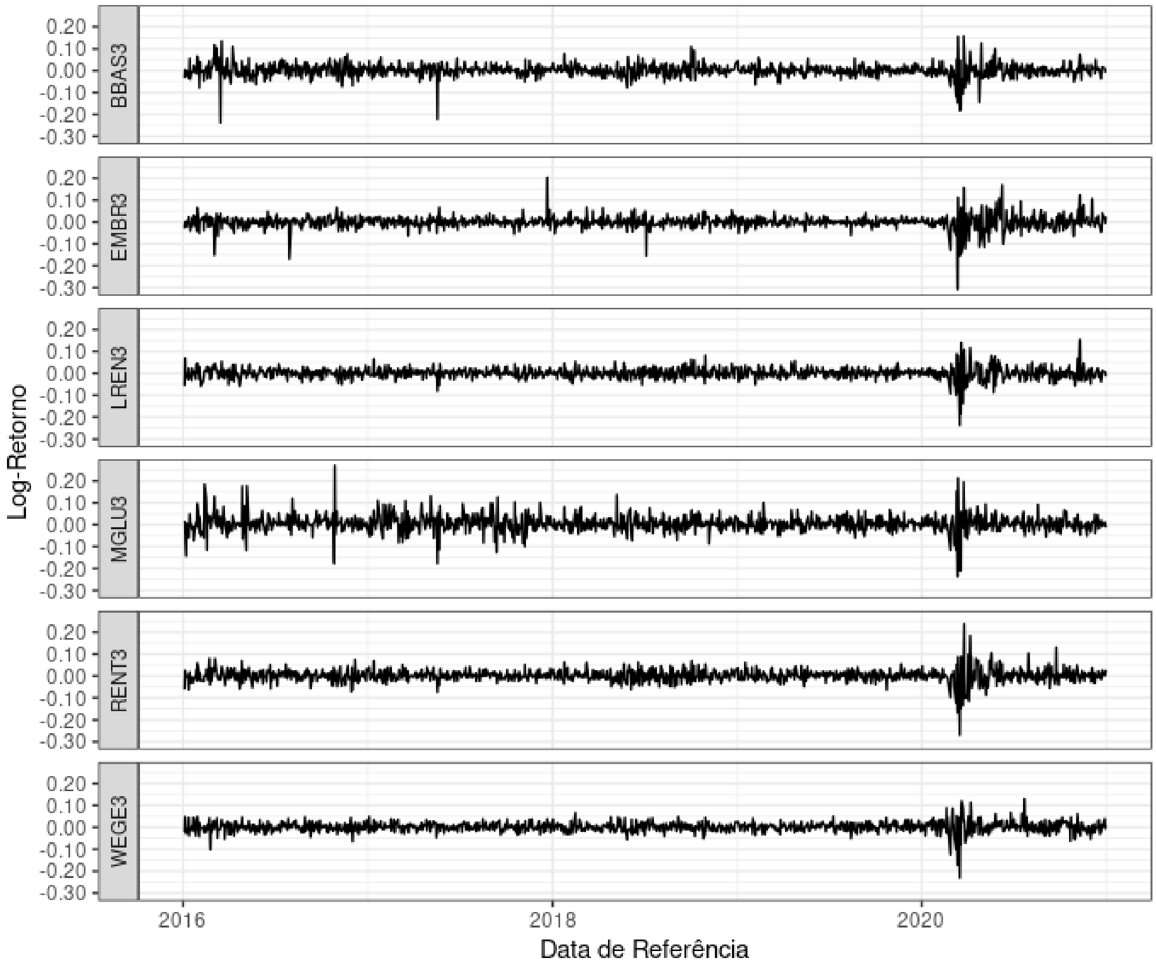

Figure 1: Log-Retornos

3.2 Métricas de Desempenho das Estratégias

A estratégia elencada na seção 2.4 foi replicada 494 vezes usando informações disponíveis em $t-1$ para estimar os pesos em t. Essa quantidade representa a quantidade de dias com negociação entre 01/01/2019 a 31/12/2020. Serão avaliados cinco estratégias para fins de comparação: (i) portfólio ingênuo $\frac{1}{N}$ com recalibragem diária, (ii) portfólio ingênuo $\frac{1}{N}$ com recalibragem mensal (primeiro dia útil de cada mês), (iii) estratégia $\text{CVaR}_{0.95}$ para determinar portfólio com *target* anual de 1%, (iv) estratégia $\text{CVaR}_{0.95}$ para determinar portfólio com *target* anual de 3% e (v) estratégia $\text{CVaR}_{0.95}$ para determinar portfólio com *target* anual de 5%.

Para cada estratégia, gerou-se um conjunto de métricas de desempenho para poder comparar diferentes características dos portfólios gerados por cada uma. Na tabela abaixo estão agrupadas as métricas

Table 2: Métricas de desempenho do portfólio

	Copula 1%	Copula 3%	Copula 5%	1/N Diário	1/N Mensal
Annualized Return	0.3546	0.3668	0.3276	0.3479	0.3344
Annualized Std Dev	0.4303	0.4302	0.4289	0.4128	0.4099
Annualized Sharpe (Rf=0%)	0.8241	0.8528	0.7639	0.8428	0.8157
Historical VaR (95%)	0.0291	0.0290	0.0292	0.0279	0.0278
Historical CVaR/ES (95%)	0.0621	0.0620	0.0614	0.0641	0.0641
Sortino Ratio (MAR = 0%)	0.0811	0.0832	0.0774	0.0802	0.0779
Downside Frequency	0.4413	0.4393	0.4453	0.4372	0.4352
Maximum Drawdown	0.4468	0.4469	0.4453	0.4905	0.4919
Average Drawdown	0.0559	0.0521	0.0509	0.0396	0.0405
Drawdown Deviation	0.0230	0.0230	0.0229	0.0237	0.0237
Average Length	20.6818	18.3600	17.8846	13.6667	14.0938

Para maior aprofundamento, ver Peterson e Carl (2020). As funções utilizadas para calcular as métricas de retorno foram implementadas no pacote *PerformanceAnalytics*. Abaixo segue uma descrição sucinta das métricas utilizadas:

1. **Annualized Return**: representa o retorno obtido por um ativo numa escala de ano. É calculado pela seguinte equação: $\sqrt[n]{prod(1+R)^{252}} - 1$, onde n é quantidade de observações e R é o vetor de retornos.

2. **Annualized Std Dev**: representa o desvio padrão do retorno (anualizado). É calculado pela seguinte equação $\sqrt{VAR(R)}\sqrt{252}$

3. **Annualized Sharpe (Rf=0%)**: representa a taxa de Sharpe anualizada, essa taxa é uma medida de retorno por risco, considera-se a taxa livre de risco igual a zero. É calculada pela seguinte equação $\frac{\sqrt[n]{prod(1+R)^{252}}-1}{\sqrt{VAR(R)}\sqrt{252}}$

4. **Historical VaR (95%)**: representa o quantil 95% da distribuição empírica dos retornos negativos. Ou seja, espera-se que a perda seja superior a esse valor em 5% dos casos.

5. **Historical CVaR/ES (95%)**: representa o valor esperado do retorno quando a perda excede o VaR.

6. **Sortino Ratio (MAR = 0%)**: Representa o risco de não atingir o *target* de retorno, definido por MAR, um acrônimo para *Minimum Acceptable Return*. Recomenda-se utilizar a taxa de retorno livre de risco como MAR. É calculada pela seguinte equação $\frac{(\overline{R-MAR})}{\delta MAR}$, onde δMAR é o *Downside Deviation* (desvio-padrão dos retornos abaixo do MAR).

7. **Downside Frequency**: representa a frequencia de retornos abaixo do mínimo retorno aceitável (0%). É dado pela equação $\frac{\Sigma_1^n I(R<0)}{n}$, onde n é quantidade total observações.

8. **Maximum Drawdown**: representa a maior perda potencial no período, que ocorre quando compra-se no ponto mais alto e vende-se no ponto mais baixo.

9. **Average Drawdown**: representa a média dos retornos negativos ao longo do período de investimento. É dado pela equação $\left|\Sigma_{j=1}^{d} \frac{D_j}{d}\right|$, onde D_j representa o j-ésimo drawdown e d a quantidade total de drawdowns no período.

10. **Drawdown Deviation**: representa a desvio-padrão dos drawdowns no período. É dado pela equação $\sqrt{\Sigma_{j=1}^{d} \frac{D_{j^2}}{n}}$, onde n é a quantidade de observações.

11. **Average Length**: representa a duração média dos drawdowns. É calculado pela equação $\Sigma_{j=1}^{d} \frac{DD_j}{d}$, onde DD_j é a quantidade de dias do j-ésimo drawdown.

A estratégia de cópulas com CVaR otimizado apresenta em todos os cenários um CVaR inferior e VaR superior à estratégia ingênua. Com exceção à cópula com retorno-alvo de 5%, observa-se que a estratégia de cópulas apresenta maiores retornos anualizados, embora com maior volatilidade que a abordagem ingênua. A cópula com alvo em 3% apresenta a melhor relação de retorno/risco pelo critério de Sharpe, seguida pela estratégia ingênua com recalibragem diária. A estratégia de cópulas apresenta sistematicamente menores perdas potenciais máximas (medido pelo *Maximum Drawdown*), embora a frequência de *downside* seja muito parecida. A estratégia de cópulas apresenta maior risco de não atingir o retorno-alvo, medido pela *Sortino Ratio*.

Em relação à riqueza do portfólio, iniciando com base 1, tem-se:

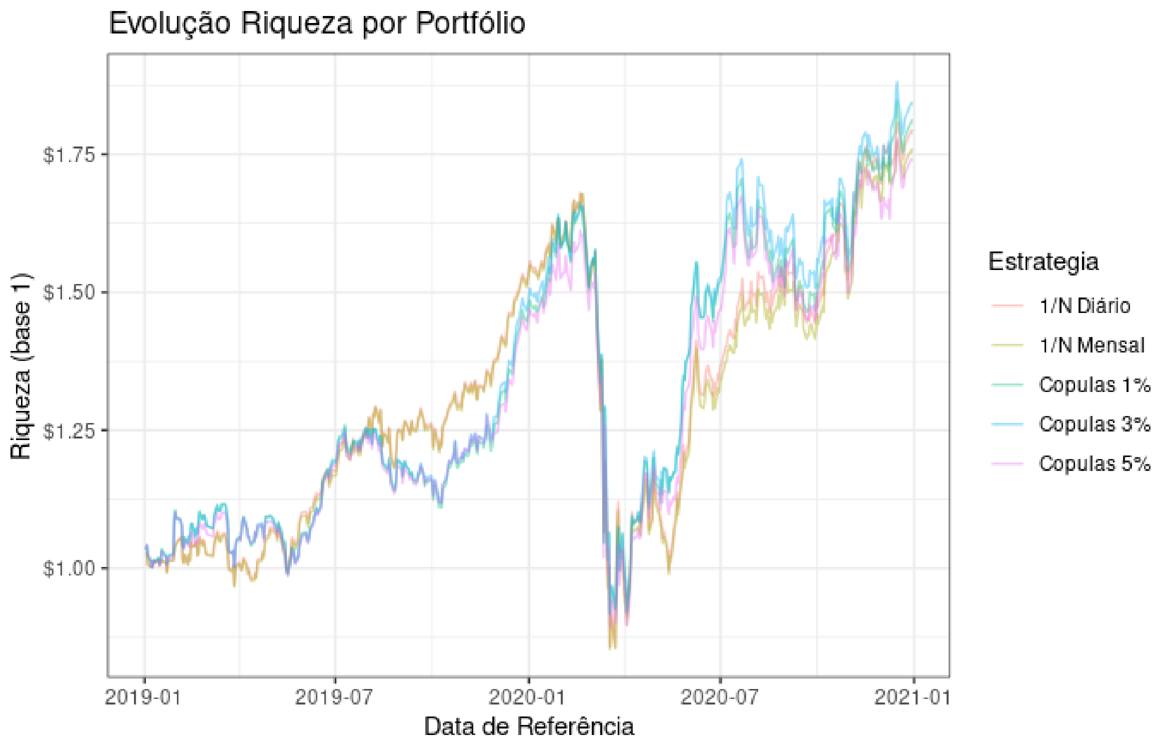

Figure 2: Evolução de Riqueza

3.3 Todos Passos

Pegando a posição com dados até 14/08/2019 para determinar os pesos para o dia 15/08/2019. Na tabela abaixo estão informações sobre a modelagem das distribuições marginais e contém os seguintes parâmetros:

- **Para a média**: ar1, ar2, ar3 e ma1

- **Para a variância**: alpha1 (arch), beta1 (garch), omega (intercepto)

- **Para as inovações t-Student assimétrica**: skew (assimetria) e shape (curtose)

Table 3: Distribuições Marginais

	BBAS3	EMBR3	LREN3	MGLU3	RENT3	WEGE3
ar1	-0.835 ***	NA	0.4829 **	NA	-0.0197	-0.0469
ar2	-0.0539 *	NA	0.0143	NA	-0.0427	0.0062
ar3	NA	NA	-0.0649 **	NA	NA	NA
ma1	0.8311 ***	NA	-0.5401 ***	NA	NA	NA
alpha1	0.107 ***	0.1606 **	0.026	0.2359 ***	0.0424 ***	0.0351 *
beta1	0.842 ***	0.5579 ***	0.9398 ***	0.4987 ***	0.9403 ***	0.917 ***
skew	1.0036 ***	1.001 ***	1.018 ***	0.9696 ***	1.0236 ***	1.0075 ***
shape	5.4235 ***	4.0768 ***	11.897 **	3.9171 ***	9.6082 ***	10.088 ***
omega	0	1e-04	0	4e-04	0	0
Q(15)	0.9955	0.6327	0.7511	0.513	0.6868	0.991
$Q^2(15)$	0.9977	0.9972	0.8982	0.9751	0.9868	0.5664
K-S Test	0	0	0	0	0	0

(***), (**), (*) indicam significância a 1%, 5% e 10%, respectivamente.

Quando está preenchido com NA, não se aplica para a série em questão.

$Q^{1,2}()$ indicam os p-valores do teste Ljung-Box até o 15º lag. A hipótese nula é de ausência de autocorrelação residual.

O K-S Test indica o p-valor do teste Kolmogorov-Smirnov para distribuição. A hipótese nula é que os dados vêm de uma distribuição uniforme-padrão $U(0, 1)$

Em relação às cópulas, observa-se que a copula t (com parâmetros $\rho = 0.3668$ e $df = 15$) é a que apresenta o menor valor para critério de informação e é, então, escolhida para representar a dependência. Todavia, observa-se que todas famílias candidatas foram rejeitadas pelo teste de qualidade de ajuste, sugerindo que não se ajustam bem aos dados.

Table 4: Métricas Cópulas

Copula	AIC	BIC	LL	GoF (p-valor)
Gumbel	-1024.9	-1019.8	513.4	0
t	-1421.4	-1411.3	712.7	0
Joe	-687.7	-682.7	344.9	0
Mixture	-1350.4	-1320.2	681.2	0

p-valor refere-se à $H_0 : C \in \mathcal{C}$

4. Conclusão

A estratégia de cópulas com CVaR otimizado apresenta em todos os cenários um CVaR inferior e VaR superior à estratégia ingênua $\frac{1}{N}$. Observa-se que a estratégia de cópulas apresenta maior volatilidade nos retornos que a abordagem ingênua. Entretanto, apresenta sistematicamente menores perdas potenciais máximas (medido pelo *Maximum Drawdown*). A estratégia de cópulas apresenta maior risco de não atingir o retorno-alvo, medido pela *Sortino Ratio.*

Próximos passos podem ser incluir mais misturas de cópulas na 'competição' para avaliar se há melhorias. Ademais, essa estratégia precisa ser replicada para um conjunto maior de ativos e de mais fontes, usando correção monetária e determinando algum ativo livre de risco como base de comparação.

5. Referências

Genest, C.; Rémillard, B.; Beaudoin, D. (2009) Goodness-of-fit tests for copulas: A review and a power study. Insurance: Mathematics and Economics 44, 199–214.

Hofert, M.; Kojadinovic, I.; Machler, M.; Yan, J. (2018) Elements of Copula Modeling with R. Springer International Publishing

Hyndman, R.J.; Khandakar, Y. (2008) Automatic Time Series Forecasting: The forecast Package for R. Journal of Statistical Software, [S.l.], v. 27, Issue 3, p. 1-22

Kahneman, D.; Tversky, A. (1979) Prospect Theory: An Analysis of Decision under Risk. Econometrica. 47 (4): 263–291

Markowitz, H. (1952) Portfolio Selection. Journal of Finance

Peterson, B.G.; Carl, P. (2020) PerformanceAnalytics: Econometric Tools for Performance and Risk Analysis. R package version 2.0.4. https://CRAN.R-project.org/package=PerformanceAnalytics

Pfaff, B. (2016) Financial Risk Modelling and Portfolio Optimization with R, Second Edition. John Wiley Sons, Ltd

Würtz, D.; Chalabi, Y.; Chen, W., Ellis, A. (2015) Portfolio Optimization with R/Rmetrics. Rmetrics Association Finance Online, www.rmetrics.org

6. Scripts

6.1 Script Geral

É o script que chama todas funções e executa os procedimentos da estratégia

```
setwd('')
# funcoes definidas pelo usu rio
source("arma_garch.R")
source("pseudo_uniform_marginal.R")
source("copulae_fit.R")
source("simulate_returns.R")
source("generate_portfolio_weights.R")
source("returns_evaluation.R")

# pacotes
library(fPortfolio)       # CUIDAR ORDEM CHAMADA DESSE PACOTE
library(PerformanceAnalytics)
library(tidyverse)
library(lubridate)
library(moments)
library(forecast)
library(rugarch)
library(copula)


# BASE DE DADOS
df <- data.table::fread('B3_stocks_long_2010-01-01_2021-04-09.csv')
df <- df %>%
  dplyr::mutate(ticker = gsub(pattern = ".SA", replacement = "", ticker)) %>%
  dplyr::filter(ticker %in% c("LREN3","RENT3","MGLU3","WEGE3","BBAS3","EMBR3")
    ) %>%
  dplyr::arrange(ref.date) %>%
  dplyr::group_by(ticker) %>%
  dplyr::mutate(
    price.dif = price.close - lag(price.close, n = 1L),
    log.return = log(price.close / lag(price.close, n = 1L))
  ) %>%
  dplyr::mutate(
    nome_acao = case_when(ticker == "LREN3" ~ "Lojas Renner",
                          ticker == "RENT3" ~ "Localiza",
                          ticker == "MGLU3" ~ "Magazine Luiza",
                          ticker == "WEGE3" ~ "WEG",
                          ticker == "BBAS3" ~ "Banco do Brasil",
                          ticker == "EMBR3" ~ "EMBRAER"
    )
  ) %>%
  dplyr::filter(year(ref.date) >= 2016 & year(ref.date) <= 2020)

# DF LOG-RETORNOS DI RIOS
df_returns <- df %>%
  dplyr::select(ref.date, ticker, log.return) %>%
  tidyr::pivot_wider(id_cols = "ref.date", names_from = "ticker", values_from
    = "log.return") %>%
  tibble::column_to_rownames("ref.date")
```

```
# SIMULAR CEN RIOS ——
  # # SIMULAR CEN RIOS: SELECIONAR PRIMEIRO DIA DE CADA SEMANA PRA
    OTIMIZA  O ——
optim_days <- df_returns %>%
  tibble::rownames_to_column(var = "ref.date") %>%
  dplyr::filter(year(ref.date) %in% c(2019, 2020)) #%>%
  # dplyr::mutate(semana = week(ref.date)) %>% dplyr::arrange(ref.date) %>%
  # dplyr::group_by(semana) %>%  dplyr::slice(1) %>% dplyr::ungroup() %>%
    dplyr::select(ref.date, semana)

    # # SIMULAR CEN RIOS: LISTA COM SIMULA  O RETORNOS T+1 ——
index_simulation <- which(row.names(df_returns) %in% optim_days$ref.date)
next_day_returns <- vector("list", length(index_simulation))
for (w in seq_along(index_simulation)) {
  x <- df_returns[seq(index_simulation[w] - 1), ]
  next_day_returns[[w]] <- simulate_returns(df_assets = x, n_simulations =
    1000)
  if (w %% 50 == 0) {
    print(w)
    print(Sys.time())
  }
}
names(next_day_returns) <- rownames(df_returns)[index_simulation - 0]

    # # SIMULAR CEN RIOS: GERAR RESULTADOS PORTF LIOS ——
real_returns <- df_returns[seq(index_simulation[1], nrow(df_returns)), ]   #
      nice   [1] porque est   ordenado
real_returns <- exp(real_returns) - 1                                         #
    converter para retornos simples

    # # SIMULAR CEN RIOS: OTIMIZAR PORTF LIOS (3 ALVOS) ——
        # # # C PULAS 1% ALVO
weights_copula_01 <- lapply(next_day_returns, function(x) { generate_portfolio
    _weights(x, target = 1.01 ** (1/252) - 1) }) %>%
  bind_rows(.id = "ref.date") %>%
  tibble::column_to_rownames("ref.date")
        # # # C PULAS 3% ALVO
weights_copula_03 <- lapply(next_day_returns, function(x) { generate_portfolio
    _weights(x, target = 1.03 ** (1/252) - 1) }) %>%
  bind_rows(.id = "ref.date") %>%
  tibble::column_to_rownames("ref.date")
        # # # C PULAS 5% ALVO
weights_copula_05 <- lapply(next_day_returns, function(x) { generate_portfolio
    _weights(x, target = 1.05 ** (1/252) - 1) }) %>%
  bind_rows(.id = "ref.date") %>%
  tibble::column_to_rownames("ref.date")
        # # # 1/N COM AJUSTES MENSAIS
first_day_month <- data.frame(dt = rownames(weights_copula_01)) %>%
  mutate(mes = month(dt), ano = year(dt)) %>% group_by(ano, mes) %>% slice(1)
   %>% ungroup()
index_first_day <- which(row.names(weights_copula_01) %in% first_day_month$dt)

weights_1n_mes <- weights_copula_01[index_first_day,]
weights_1n_mes[seq(nrow(weights_1n_mes)),] <- 1 / ncol(weights_1n_mes)
```

```
        ### 1/N COM AJUSTES DI RIOS
weights_1n_dia <- weights_copula_01
weights_1n_dia[seq(nrow(weights_copula_01)),] <- 1 / ncol(weights_copula_01)



    ## SIMULAR CEN RIOS: GERAR M TRICAS DE DESEMPENHO ----
portfolio_copula_01 <- returns_evaluation(weights_portfolio = weights_copula_
    01,
                                          actual_returns = real_returns)
portfolio_copula_03 <- returns_evaluation(weights_portfolio = weights_copula_
    03,
                                          actual_returns = real_returns)
portfolio_copula_05 <- returns_evaluation(weights_portfolio = weights_copula_
    05,
                                          actual_returns = real_returns)
portfolio_1n_mes <- returns_evaluation(weights_portfolio = weights_1n_mes,
                                       actual_returns = real_returns)
portfolio_1n_dia <- returns_evaluation(weights_portfolio = weights_1n_dia,
                                       actual_returns = real_returns)

# COMPARAR RESULTADOS ----
    ## COMPARAR RESULTADOS: RETORNOS ----
list({portfolio_copula_01$metrics %>% rename('Copula 1%' = wealth)},
     {portfolio_copula_03$metrics %>% rename('Copula 3%' = wealth)},
     {portfolio_copula_05$metrics %>% rename('Copula 5%' = wealth)},
     {portfolio_1n_dia$metrics %>% rename('1/N Di rio' = wealth)},
     {portfolio_1n_mes$metrics %>% rename('1/N Mensal' = wealth)}) %>%
  purrr::reduce(full_join, by = "metric") %>%
  knitr::kable(., "latex", digits = 4)


    ## COMPARAR RESULTADOS: PLOT RIQUEZA ----
list(data.frame(Estrategia = "Copulas 1%", portfolio_copula_01$wealth) %>%
    tibble::rownames_to_column("ref.date"),
     data.frame(Estrategia = "Copulas 3%", portfolio_copula_03$wealth) %>%
    tibble::rownames_to_column("ref.date"),
     data.frame(Estrategia = "Copulas 5%", portfolio_copula_05$wealth) %>%
    tibble::rownames_to_column("ref.date"),
     data.frame(Estrategia = "1/N Mensal", portfolio_1n_mes$wealth) %>% tibble
    ::rownames_to_column("ref.date"),
     data.frame(Estrategia = "1/N Di rio", portfolio_1n_dia$wealth) %>%
    tibble::rownames_to_column("ref.date")
     ) %>%
  bind_rows() %>%
  mutate(ref.date = ymd(ref.date)) %>%
  ggplot(., aes(x = ref.date, y = wealth, colour = Estrategia)) +
    geom_line(alpha = 0.4) +
    scale_y_continuous(labels = scales::dollar_format()) +
    labs(y = "Riqueza (base 1)", x = "Data de Refer ncia", title = "
    Evolu  o Riqueza por Portf lio") +
    theme_bw()
```

6.2 ARMA-GARCH

É a função que estima o modelo ARMA-GARCH com inovações t-student assimétrica

```
# ETAPA ARMA-GARCH
    # ESTIMAR ARMA SEM  M DIA  E SEM SAZONALIDADE (RETORNOS SEMPRE  M DIA  0)
    # USAR ORDEM SELECIONADA PARA ESTIMAR GARCH
      # MODELO VARI NCIA     STANDARD GARCH
      # MODELO  M DIA  USA ORDENS DO ARMA ESTIMADO
      # INOVA  ES SEGUEM T-SUDENT SKEWED
    # RETORNA OBJETO GARCH

arma_garch <- function(y) {
  # ESTIMATE ARMA MODEL
  arima_model <- forecast::auto.arima(y = y,
                                      stationary = TRUE,
                                      seasonal = FALSE,
                                      allowmean = FALSE,
                                      allowdrift = FALSE
  )

  arima_order <- forecast::arimaorder(arima_model)[c(1,3)]

  # ESTIMATE GARCH MODEL
  garch_specs <- rugarch::ugarchspec(
    variance.model = list(model = "sGARCH",        # STANDARD GARCH
                          garchOrder = c(1,1),     # c(0,1) FOR ARCH
                          submodel = NULL,         # VALID FOR model = 'fGARCH
                          variance.targeting = T  # VARIANCE TARGETING FOR VAR
    INTERCEPT OMEGA (CARE WITH  fGARCH)
    ),
    mean.model = list(armaOrder = arima_order,    # ORDER FOR ARIMA (
   PREVIOUSLY CALCULATED)
                      include.mean = F,            # RETURNS MEAN EQUAL TO ZERO
                      archm = F,                   # ARCH VOLATILITY ON MEAN (
   VALID FOR include.mean = T)
                      archpow = 1                  # SD OR VAR FOR MEAN
   REGRESSION (VALID FOR include.mean = T)
    ),
    distribution.model = 'sstd'                    # INNOVATIONS (SKEWED
   STUDENT (GIVEN STYLIZED FACTS))
  )

  garch_model <- rugarch::ugarchfit(spec = garch_specs,
                                    data = y,
                                    solver = "solnp",
                                    solver.control = list(n.restarts = 2),
                                    out.sample = 0  # CAN KEEP OUT-OF-SAMPLE
   TO EVALUATE RESULTS
  )
  return(garch_model)
}
```

6.3 Pseudo Uniform Marginals

É a função que cria os vetores das pseudo-marginais a partir dos parâmetros do GARCH

```
# ETAPA CRIAR CDF DAS MARGINAIS
    # A PARTIR DE UM OBJETO DO MODELO GARCH
    # ESTIMA AS OBSERVA ES PSEUDO-UNIFORMES
    # SE skewed_t = TRUE, USA SKEWED T-STUDENT
    # SE skewed_t = FALSE, USA FUN  O POBS DO PACOTE COPULA

pseudo_uniform_marginal <- function(garch_model, skewed_t = TRUE) {
  # VECTORS
  errors_vector <- residuals(garch_model)
  sigma_vector <- garch_model@fit$sigma
  coefficients_vector <- garch_model@fit$coef

  if (skewed_t) {
    coefs_df <- tibble::enframe(coefficients_vector)
    uniform_vector <- fGarch::psstd(q = errors_vector / sigma_vector,
                                     nu = coefs_df$value[coefs_df$name == '
    shape'],
                                     xi = coefs_df$value[coefs_df$name == 'skew
    '])
  } else {
    uniform_vector <- copula::pobs(errors_vector / sigma_vector)
  }
  return(uniform_vector)
}
```

6.4 Fit Copula

É a função que compara as cópulas candidatas e retorna a com menor AIC as métricas de comparação entre as cópulas (opcionalmente retorna o teste de qualidade de ajuste)

```
# COPULAS
    # A PARTIR DE UM DATAFRAME COM LOG.RETORNO DE ATIVOS & PSEUDO_OBSERVATIONS
    # ESTIMA C PULAS:
      # EL PTICAS: t
      # ARQUIMEDIANAS: gumbel, Joe
      # DE VALOR EXTREMO: gumbel

# library(copula)
copulae_fit <- function(x_df, pseudo_observations, goft = FALSE) {
  dimension <- ncol(x_df)
  # ESTIMAR COPULAS ----
      # # ESTIMAR COPULAS: GUMBEL
  copulaGumbel <- fitCopula(copula = gumbelCopula(dim = dimension),
                            data = pseudo_observations, method = "mpl")

      # # ESTIMAR COPULAS: t
  copulaT <- fitCopula(copula = tCopula(dim = dimension,
                                        dispstr = "ex",
                                        df.fixed = FALSE),
                       data = pseudo_observations, method = "mpl")

      # # ESTIMAR COPULAS: JOE
  copulaJoe <- fitCopula(copula = joeCopula(dim = dimension),
                         data = pseudo_observations, method = "mpl")

      # # ESTIMAR COPULAS: MISTA
  copulaMix <- fitCopula(copula = mixCopula(list(gumbelCopula(dim = dimension)
    ,
                                                 # rotCopula(gumbelCopula(dim
   = dimension)),
                                                 tCopula(dim = dimension, df.
    fixed = T),
                                                 joeCopula(dim = dimension))),
                         data = pseudo_observations, method = "mpl")

  # M TRICAS PADR O C PULAS ----
  copulae_metrics <- data.frame(
    Copula = c("Gumbel", "t", "Joe", "Mixture"),
    AIC = c(AIC(copulaGumbel), AIC(copulaT), AIC(copulaJoe), AIC(copulaMix)),
    BIC = c(BIC(copulaGumbel), BIC(copulaT), BIC(copulaJoe), BIC(copulaMix)),
    LL = c(logLik(copulaGumbel), logLik(copulaT), logLik(copulaJoe), logLik(
   copulaMix))
  )

  # REALIZA TESTE DE QUALIDADE DE AJUSTE? (MUITO DEVAGAR)
  if (goft) {
    # QUALIDADE DO AJUSTE
    gofGumbel <- gofCopula(copula = copulaGumbel@copula, x = x_df, N = 200,
   method = "Sn", simulation = c("pb", "mult")[1])
    gofJoe <- gofCopula(copula = copulaJoe@copula, x = x_df, N = 200, method =
    "Sn", simulation = c("pb", "mult")[1])
```

```
    ### AP S ESTIMAR DF, CALCULAR COMO DF.FIXED PARA AGILIZAR TESTE
   QUALIDADE DE AJUSTE (t e mista)
        #### COPULA T
    degrees_freedom <- as.integer(round(copulaT@copula@parameters[2], digits =
     0))
    copulaTT <- fitCopula(copula = tCopula(dim = dimension, dispstr = "ex", df
     = degrees_freedom, df.fixed = TRUE),
                             data = pseudo_observations,
                             method = "mpl")
    gofT <- copula::gofCopula(copula = copulaTT@copula, x = x_df, N = 50,
   method = "Sn", simulation = c("pb", "mult")[1])

        #### COPULA MISTA (T NA POSI  O 2)
    degrees_freedom_mista <- as.integer(round( copulaMix@copula@cops@.Data
    [[2]]@parameters[2], digits = 0))
    copulaMixx <- fitCopula(copula = mixCopula(list(gumbelCopula(dim =
   dimension),
                                                    # rotCopula(gumbelCopula(
   dim = dimension)),
                                                    tCopula(dim = dimension,
   df.fixed = T, df = degrees_freedom_mista),
                                                    joeCopula(dim = dimension)
   )),
                              data = pseudo_observations, method = "mpl")

    gofMix <- gofCopula(copula = copulaMixx@copula, x = x_df, N = 50, method =
     "Sn", simulation = c("pb", "mult")[1])

    # COMPILAR M TRICAS
    copulae_metrics <- bind_cols(list(copulae_metrics,
                                      data.frame(GoF = c(gofGumbel$p.value,
   gofT$p.value,
                                                         gofJoe$p.value,
   gofMix$p.value)
                                      )))
  }

  # VENCEDORA
  index_best <- which.min(copulae_metrics$AIC)
  copulaFinal <- list(copulaGumbel, copulaT, copulaJoe, copulaMix)[[index_best
    ]]

  # RETORNAR
  result <- list(final_copula = copulaFinal, metricas = copulae_metrics)
return(result)
}
```

6.5 Simular Retornos

É a função que simula os retornos do período seguinte a partir do modelo ARMA-GARCH e da cópula

```
simulate_returns <- function(df_assets, n_simulations = 500) {
  n_assets <- ncol(df_assets)

  ## ARMA_GARCH + PSEUDO_OBSERVATIONS ----
  garch_models <- lapply(df_assets, function(x) {arma_garch(y = x)})
  pseudo_obs <- lapply(garch_models, function(x) {pseudo_uniform_marginal(
    garch_model = x, skewed_t = TRUE)})
  pseudo_obs_matrix <- pseudo_obs %>% bind_cols() %>% as.matrix()

  df_std_resid <- lapply(garch_models, function(x) {residuals(x) / x@fit$sigma
    }) %>% bind_cols() %>% as.matrix()

  ## ESTIMAR COPULA
  copula_output <- copulae_fit(x_df = df_std_resid, pseudo_observations =
    pseudo_obs_matrix)
  final_copula <- copula_output$final_copula

  ## SIMULAR RETORNOS PARA T+1
        ## gerar valores c pulas
  copulae_simulate <- copula::rCopula(n = n_simulations, copula = final_
    copula@copula)
        ## previsao + simulacao
  returns_simulation <- mapply(function(garch, copulae) {
          ### converter "pseudo" gerado pelas c pulas para estimativas
    de erro
    garch_coefs <- as.data.frame(t(coef(garch)))
    q_garch <- fGarch::qsstd(p = copulae,
                             nu = garch_coefs$shape,
                             xi = garch_coefs$skew)
    z_sim <- q_garch / sd(q_garch)
          ### prever s rie e sigma
    garch_forecast <- rugarch::ugarchforecast(garch, n.ahead = 1)

          ### retorno
    retornos <- as.vector(garch_forecast@forecast$seriesFor) + as.vector(garch
    _forecast@forecast$sigmaFor) * z_sim

  }, garch = garch_models, copulae = as.data.frame(copulae_simulate))

return(returns_simulation)
}
```

6.6 Gerar Pesos do Portfólio

É a função que determina o peso de cada ativo no portfólio a partir dos retornos simulados (otimização do CVaR)

```
generate_portfolio_weights <- function(next_day_returns, target) {
  # ESPECIFICA  ES PORTFOLIO
  portfolio_spec <- fPortfolio::portfolioSpec(
    model = list(type = "CVaR",
                 params = list(alpha = 0.05),
                 tailRisk = list()),
    portfolio = list(
      targetReturn = target
    ),
    optim = list(
      solver = "solveRglpk.CVAR"
    )
  )
  # OTIMIZA  O PORTFOLIO
  portfolio_return <- as.timeSeries(next_day_returns)
  portfolio_final <- fPortfolio::efficientPortfolio(data = portfolio_return,
                                                    spec = portfolio_spec)
  # PESOS DO PORTF LIO
  portfolio_weights <- portfolio_final@portfolio@portfolio$weights
  portfolio_weights <- data.frame(t(portfolio_weights / sum(portfolio_weights)
    ))

return(portfolio_weights)
}
```

6.7 Avaliar Retornos

É a função que traz métricas para comparar o resultado fora-da-amostra de cada estratégia

```
returns_evaluation <- function(weights_portfolio, actual_returns, colname = "
    wealth") {
  # RETURNS
  portfolio_returns <- PerformanceAnalytics::Return.portfolio(R = actual_
    returns,
                                                              weights =
    weights_portfolio,
                                                              wealth.index = F
    , value = 1)
  # WEALTH
  portfolio_wealth <- PerformanceAnalytics::Return.portfolio(R = actual_
    returns,
                                                             weights = weights
    _portfolio,
                                                             wealth.index = T,
     value = 1)
  # PERFORMANCE METRICS
  portfolio_metrics <- list(
    table.AnnualizedReturns(portfolio_returns),
    table.DownsideRisk(portfolio_returns)[8:9, , drop = FALSE],
    # SharpeRatio(portfolio_returns, method = "historical"),
    SortinoRatio(portfolio_returns, method = "historical"),
    data.frame(portfolio.returns = DownsideFrequency(portfolio_returns), row.
   names = "Downside Frequency"),
    data.frame(portfolio.returns = maxDrawdown(portfolio_returns), row.names =
     "Maximum Drawdown"),
    AverageDrawdown(portfolio_returns, method = "historical"),
    DrawdownDeviation(portfolio_returns),
    AverageLength(portfolio_returns, method = "historical")
  ) %>%
    lapply(., function(x) {data.frame(x) %>% tibble::rownames_to_column("
    metric")}) %>%
    bind_rows()

  # RENAMES COLUMNS
  colnames(portfolio_returns) <- colname
  colnames(portfolio_wealth) <- colname
  colnames(portfolio_metrics)[2] <- colname

  # RESULT
  resultado <- list(
    returns = portfolio_returns,
    wealth = portfolio_wealth,
    metrics = portfolio_metrics
  )

return(resultado)
}
```